

Scope ambiguities, monads and strengths

Justyna Grudzińska

Marek Zawadowski

August 2, 2016

Abstract

In this paper, we will discuss three semantically distinct scope assignment strategies: traditional movement strategy ([16], [19], [6]), polyadic approach ([17], [9], [10], [22], [5]), and continuation-based approach ([1], [7], [2], [11], [4]). As a generalized quantifier on a set X is an element of $\mathcal{C}(X)$, the value of the continuation monad \mathcal{C} on X , in all three approaches QPs are interpreted as \mathcal{C} -computations. The main goal of this paper is to relate the three strategies to the computational machinery connected to the monad \mathcal{C} (strength and derived operations). As will be shown, both the polyadic approach and the continuation-based approach make heavy use of monad constructs. In the traditional movement strategy, monad constructs are not used but we still need them to explain how the three strategies are related and what can be expected of them wrt handling scopal ambiguities in simple sentences.

1 Scope ambiguities

Multi-quantifier sentences have been known to be ambiguous with different readings corresponding to how various quantifier phrases (QPs) are semantically related in the sentence. For example,

- (1) Every girl likes a boy

admits of the subject wide scope reading ($S > O$) where each girl likes a potentially different boy, and the object wide scope reading ($O > S$) where there is one boy whom all the girls like. As the number of QPs in a sentence increases, the number of distinct readings also increases. Thus a simple sentence with three QPs admits of six possible readings, and in general a simple sentence with n QPs will be (at least) $n!$ ways ambiguous (we only consider readings where QPs are linearly ordered - what we will call asymmetric readings).

In this paper, we will discuss three semantically distinct scope assignment strategies

Strategy A: Traditional movement strategy ([16], [19], [6]).

Strategy B: Polyadic approach ([17], [9], [10], [22], [5]).

Strategy C: Continuation-based approach ([1], [7], [2], [11], [4]).

Scope assignment strategies can be divided into two families: movement analyses (Strategies A and B) and in situ analyses (Strategy C). Strategy A has been implemented in various ways using May’s QR ([16]), Montague’s Quantifying In Rule ([19]), Cooper’s Storage ([6]). Strategy B involving polyadic quantification has been first introduced in the works of May ([17]), Keenan ([9]), Zawadowski ([22]) and van Benthem ([5]). The most recent Strategy C involves continuations and has been first proposed in the works of Barker ([1]) and de Groote ([7]), and then further developed and modified in the works of Barker and Shan ([2]), Kiselyov and Shan ([11]) and Bekki and Asai ([4]). The continuation-based strategies can be divided into two groups: those that locate the source of scope-ambiguity in the rules of semantic composition and those that attribute it to the lexical entries for the quantifier words. In this paper, we only consider operation-based approaches (as in [1]). As a generalized quantifier on a set X is an element of $\mathcal{C}(X)$, the value of the continuation monad \mathcal{C} on X , in all three approaches QPs are interpreted as \mathcal{C} -computations. The main goal of this paper is to relate the three scope assignment strategies to the computational machinery connected to the monad \mathcal{C} (strength and derived operations). As will be shown, Strategies B and C make heavy use of monad constructs. In Strategy A, monad constructs are not used but we still need them to explain how the three strategies are related and what can be expected of them wrt handling scopal ambiguities in simple sentences.

2 Monads and strenghts

For unexplained notions related to category theory, we refer the reader to standard textbooks on category theory. We shall be exclusively working in the cartesian closed category of sets *Set*. The category *Set* of sets has sets as objects. A morphism in *Set* from an object (set) X to an object (set) Y is a function $f : X \rightarrow Y$ from X to Y .

2.1 Monads

A *monad* on Set is a triple (T, η, μ) where $T : Set \rightarrow Set$ is an endofunctor (the underlying functor of the monad), $\eta : 1_{Set} \rightarrow T$ and $\mu : T^2 \rightarrow T$ are natural transformations (first from identity functor on Set to T , second from the composition of T with itself to T) making the following diagrams

$$\begin{array}{ccc}
 T & \xrightarrow{\eta_T} & T^2 & \xleftarrow{T(\eta)} & T \\
 & \searrow 1_T & \downarrow \mu & & \swarrow 1_T \\
 & & T & &
 \end{array}
 \qquad
 \begin{array}{ccc}
 T^3 & \xrightarrow{\mu_T} & T^2 \\
 T(\mu) \downarrow & & \downarrow \mu \\
 T^2 & \xrightarrow{\mu} & T
 \end{array}$$

commute. η and μ are often referred to as *unit* and *multiplication* of the monad T , respectively. These diagrams express the essence of the algebraic calculations. We shall explain their meaning while describing the list monad below.

Monads can serve many different purposes. Here, we think of a monad as a device to extend the notion of computation. We think of a function $f : X \rightarrow Y$ as a computation that, when given an element of x , provides (computes) an element $f(x)$ of Y . Then the function $f : X \rightarrow T(Y)$ can be thought of as a computation that, when given an element x in X , provides a computation in $T(Y)$ that might, in principle, evaluate to an element of Y . We shall illustrate the concept on some examples below, before we focus on the continuation monad - the main notion of computation considered in this paper.

Examples of monads.

1. *Identity monad* is the simplest possible monad but not very interesting. In this case the functor T and the natural transformations η and μ are identities. For this monad, the notion of a T -computation in X is just an element of X , as the function $f : X \rightarrow T(Y)$ is just $f : X \rightarrow Y$.
2. *Maybe monad* is the simplest non-trivial monad. The functor T associates to every set X the set $T(X) = X + \{\perp\}$ (the disjoint sum of X and singleton $\{\perp\}$), and to every function $f : X \rightarrow Y$ a function $T(f) : T(X) \rightarrow T(Y)$ such that, for $x \in T(X)$,

$$T(f)(x) = \begin{cases} x & \text{if } x \in X \\ \perp & \text{if } x = \perp \end{cases}$$

So T adds to X an additional element \perp , called *bottom* or *nothing*. The component at X of natural transformation η is a function $\eta_X : X \rightarrow X + \{\perp\}$ such that $\eta_X(x) = x$, i.e. it sends x to the same x but in the set $X + \{\perp\}$. The component at X of natural transformation μ is a function $\mu_X : X + \{\perp, \perp'\} \rightarrow X + \{\perp\}$ such that, for $x \in X + \{\perp, \perp'\}$,

$$\mu_X(x) = \begin{cases} x & \text{if } x \in X \\ \perp & \text{if } x = \perp \text{ or } x = \perp' \end{cases}$$

i.e. it sends x in X to the same x , and two bottoms \perp and \perp' in $T^2(X)$ to the only bottom \perp in $T(X)$.

For this monad, the notion of a T -computation in X consists of elements of X and an additional computation \perp that says that we do not get a value in X . The function $f : X \rightarrow T(Y)$ is just a partial function $f : X \rightarrow Y$. So this monad allows to treat partial computations as total.

3. *Exception monad* is still less trivial than maybe monad. We are given a fixed set of exceptions E and, for a set X , the monad functor is $T(X) = X + E$, i.e. the disjoint union of X and E . If E is empty, it is the identity monad; if E is a singleton, then it is a maybe monad; otherwise is it like maybe monad but with many options for nothingness.
4. *List monad* or *monoid monad* is still more interesting than the previous monad and we shall work it out in detail. It is not needed for the applications in the paper but it provides some intuitions before we move to the continuation monad. To any set X the list monad functor associates the set $T(X)$ of (finite) words over X (treated as an alphabet). This includes the empty word ε . To a function $f : X \rightarrow Y$ the functor T associates the function $T(f) : T(X) \rightarrow T(Y)$ sending the word x_1, x_2, \dots, x_n over X to the word $f(x_1), f(x_2), \dots, f(x_n)$ over Y . The component at X of natural transformation η is a function $\eta_X : X \rightarrow T(X)$ such that $\eta_X(x) = x$, i.e. it sends (the letter) x to the one letter word x in $T(X)$.

The component at X of natural transformation μ is a function $\mu_X : T^2(X) \rightarrow T(X)$. Note that $T^2(X) = T(T(X))$ is the set of words whose letters are words over the alphabet X . Thus it can be thought of as a list of lists. μ_X applied to such a list of lists flattens it to the single list. A three letter word $t = (x_1, x_2), (x_3, x_4, x_5), \varepsilon$ is a typical element of $T^2(X)$. The result of flattening T is the list

$\mu_X(T) = x_1, x_2, x_3, x_4, x_5$ in $T(X)$. We can think of such a word w as a term/word/computation $u = y_1, y_2, y_3$ in which we intend to substitute the term $v_1 = x_1, x_2$ for variable y_1 , the term $v_2 = x_3, x_4, x_5$ for variable y_2 , and the term $v_3 = \varepsilon$ for variable y_3 , i.e. $u[y_1 \setminus v_1, y_2 \setminus v_2, y_3 \setminus v_3]$. Now the multiplication μ can be thought of as an actual substitution. With this interpretation one can understand the intuitions behind the monad diagrams. In the left triangle, an element of $T(X)$, say x_1, x_2, x_3 , is mapped through $\eta_{T(X)}$ to single letter word (x_1, x_2, x_3) and μ_X flattens it back to x_1, x_2, x_3 , as required for the triangle to commute. In other words, the substitution $y[y \setminus v]$ results in v . In the right triangle, the map $T(\eta_x)$ sends, say x_1, x_2, x_3 , to the letter word $(x_1), (x_2), (x_2)$ with each letter being a single letter word. Thus again flattening such a list gives x_1, x_2, x_3 back, as required. In other words the substitution $y_1, y_2, y_3[y_1 \setminus x_1, y_1 \setminus x_2, y_3 \setminus x_3]$ results in x_1, x_2, x_3 . Thus the above triangles ensure that if we substitute with either term being a variable, then we get the expected result. The commutation of the square diagram, in this case, expresses the fact if we have a list of lists of lists and we flatten it in two different ways, once starting with the upper two levels of lists and the other time starting with the lower two levels of lists, and then we flatten the results again to get the ordinary lists over X in $T(X)$, these lists coincide. On a more conceptual level, this square expresses the fact that the evaluation commutes with substitution. In this sense these diagrams capture the essence of all algebraic calculations.

For this monad, the notion of a T -computation in X consists of words over X to be evaluated/computed in a monoid when elements of X will be (interpreted) in a monoid. The function $f : X \longrightarrow T(Y)$ is just a function $f : X \longrightarrow T(Y)$ sending elements of X to words over Y . So this monad allows for a list of values, for a given input.

5. See next subsection for unexplained notation.

(Covariant) power-set monad sends set X to power-set $\mathcal{P}(X)$ and a function $f : X \longrightarrow Y$ to the image function $\mathcal{P}(f) = \vec{f} : \mathcal{P}(X) \longrightarrow \mathcal{P}(Y)$ such that, for $h : X \rightarrow \mathbf{t}$, $\vec{f}(h)$ is an image of U under the function f , i.e. for $y \in Y$

$$\vec{f}(h)(y) = \bigvee_{x \in X, f(x)=y} h(x).$$

The unit $\eta_X : X \longrightarrow \mathcal{P}(X)$ embeds x in X to the characteristic func-

tion x , i.e.

$$\eta_X(x)(x') = \begin{cases} \mathbf{true} & \text{if } x = x' \\ \mathbf{false} & \text{otherwise.} \end{cases}$$

The multiplication $\mu_X : \mathcal{P}^2(X) \rightarrow \mathcal{P}(X)$ sums elements of elements of elements, i.e. for $H \in \mathcal{P}^2(X)$ and $x \in X$, $\mu_X(H)(x) : X \rightarrow \mathbf{t}$ is a function given by

$$\mu_X(H)(x) = \bigvee_{h \in \mathcal{P}(X)} H(h) \wedge h(x).$$

This monad also allows for a set of values, for a given input.

2.2 Notation

Before we explain the notion of computation coming with the continuation monad, we restate the monad in a more functional way. To do this, we need to introduce some notation. As *Set* is a cartesian closed category, it is customary to denote functions between sets using λ notation. One can think of it as if we were to work in the internal language of *Set*, i.e. λ theory where all functions have their names represented. For sets X and Y , we shall use $X \times Y$ to denote the binary product of X and Y and $X \Rightarrow Y$ to denote the set of functions from X to Y . As it is customary, we associate \Rightarrow to the right, i.e. $X \Rightarrow Y \Rightarrow Z$ means $X \Rightarrow (Y \Rightarrow Z)$ and this set is naturally bijective with $(X \times Y) \Rightarrow Z$. If we have a function

$$f : X \times Y \longrightarrow Z,$$

then by

$$\lambda y.Y.f : X \longrightarrow Y \Rightarrow Z$$

we denote its exponential adjunction, i.e. the function from X to the set of functions $Y \Rightarrow Z$ such that, for an element $x \in X$, $\lambda y.Y.f(x)$ is a function from Y to Z such that, for an element $y \in Y$, $(\lambda y.Y.f)(x)(y)$ is by definition equal $f(x, y)$. Note that in the expression $(\lambda y.Y.f)(x)(y)$ the first occurrence of y is an occurrence of a variable (as it is part of the name of a function), whereas the second occurrence of y in this expression denotes an element of the set Y .

π_i will denote the projection on i -component from the product. Any function $\sigma : \{1, \dots, m\} \rightarrow \{1, \dots, n\}$ induces a generalized projection denoted

$$\pi_\sigma = \langle \pi_{\sigma(1)}, \dots, \pi_{\sigma(m)} \rangle : X_1 \times \dots \times X_n \longrightarrow X_1 \times \dots \times X_m.$$

We will use this notation mainly for σ 's being bijections, i.e. when π_σ is just permutation of the component for the product.

We have a distinguished set of truth values $\mathbf{t} = \{\mathbf{true}, \mathbf{false}\}$. We shall use the usual (possibly infinitary) operations on this set. For a set X , we put $\mathcal{P}(X) = X \Rightarrow \mathbf{t}$, i.e. the (functional) powerset of X .

2.3 Continuation monad

Continuation monad, the most important for us, denoted \mathcal{C} , has some similarities to the power-set monad but it also differs in a substantial way. At the level of objects, it is just twice iterated power-set construction, i.e. for set X , $\mathcal{C}(X) = \mathcal{P}^2(X)$, but at the level of morphisms, it is an inverse image of an inverse image, i.e., function $f : X \rightarrow Y$ induces an inverse image function between powersets

$$\mathcal{P}(f) = f^{-1} : \mathcal{P}(Y) \rightarrow \mathcal{P}(X)$$

$$h \mapsto h \circ f, \quad \mathcal{P}(f) = \lambda h : \mathcal{P}(Y). \lambda x : X. h(f x)$$

Taking again an inverse image function, we have

$$\mathcal{C}(f) = \mathcal{P}(f^{-1}) : \mathcal{C}(X) \rightarrow \mathcal{C}(Y)$$

$$Q \mapsto Q \circ f^{-1}, \quad \mathcal{C}(f)(Q) = \lambda h : \mathcal{P}(Y). Q(\lambda x : X. h(f x))$$

for $Q \in \mathcal{C}(X)$.

The unit $\eta_X : X \rightarrow \mathcal{C}(X)$ is given by

$$\eta_X(x) = \lambda h : \mathcal{P}(X). h(x).$$

for $x \in X$.

The multiplication $\mu_X : \mathcal{C}^2(X) \rightarrow \mathcal{C}(X)$ can be explained in terms of η

$$\mu_X = (\eta_{\mathcal{P}(X)})^{-1} : \mathcal{P}^4(X) \rightarrow \mathcal{P}^2(X).$$

In other words, $\mu_X(\mathcal{F}) : \mathcal{P}(X) \rightarrow \mathbf{t}$ is a function such that

$$\mu_X(\mathcal{F})(h) = \mathcal{F}(\eta_{\mathcal{P}(X)}(h))$$

for $\mathcal{F} : \mathcal{P}^3(X) \rightarrow \mathbf{t}$ and $h : X \rightarrow \mathbf{t}$.

In λ -notation, we write

$$\mu_X(\mathcal{F})(h) = \mathcal{F}(\lambda D : \mathcal{C}(X). D(h)).$$

Now we can look at the notion of computation related the continuation monad. Consider the function

$$f : X \longrightarrow \mathcal{C}(Y).$$

By exponential adjunction (uncurrying) it corresponds to a function

$$f' : \mathcal{P}(Y) \times X \longrightarrow \mathbf{t}$$

and again by exponential adjunction (currying) it corresponds to a function

$$f'' : \mathcal{P}(Y) \longrightarrow \mathcal{P}(X).$$

Thus a \mathcal{C} -computation from X to Y is a function that sends functions from $\mathcal{P}(Y) = Y \Rightarrow \mathbf{t}$ to functions in $\mathcal{P}(X)$. So instead of having a direct answer for a given element $x \in X$ what is the value $f(x)$ in Y , we are given for every continuation function $c : Y \longrightarrow \mathbf{t}$ a value in the answer type \mathbf{t} that could be thought of as $c(f(x))$ (if there were an element in Y that could be reasonably called $f(x)$). We can draw the picture illustrating the situation

$$X \xrightarrow{f?} Y \xrightarrow{c} \mathbf{t}$$

$f(c)$

Instead of ‘procedure’ $f?$ computing y ’s from x ’s (that we don’t have), we provide a continuation $f(c)$ for any continuation (of the computation) c . If $f?$ would be indeed a genuine function $f? : X \rightarrow Y$, then $f(c)$ would be the composition $c \circ f?$.

2.4 Bi-strong monads

As the notion of strength is new in this context, we shall briefly recall its history. There are three manifestations of strength on a functor. Historically, the first one was the notion of enrichment of a functor (c.f. [8]). The tensorial strength (i.e., natural transformation of a kind $X \otimes T(Y) \rightarrow T(X \otimes Y)$ used in this paper) was introduced in [12] and further developed in [14]. The cotensorial strength (i.e., natural transformation of a kind $T(X \Rightarrow Y) \rightarrow X \Rightarrow T(Y)$) introduced in [13] also proved useful in some contexts. In symmetric monoidal closed categories these concepts are equivalent, (c.f. [13]).

As it was noticed in [18], a monad, in order to have a well behaved notion of computation, has to be strong. Fortunately, all monads on Set are strong. More precisely, all monads on set can be canonically equipped with two

strengths, left and right, and moreover these strengths are compatible in a precise technical sense. This additional structure on the continuation monad will be essential when we shall analyze the meaning of multiple quantified sentences.

Let (T, η, μ) be a monad on Set . The *left strength* is a natural transformation with components

$$\mathbf{st}^l_{X,Y} : T(X) \times Y \longrightarrow T(X \times Y)$$

for sets X and Y , making the diagrams

$$\begin{array}{ccc} T(X) \times Y \times Z & \xrightarrow{\mathbf{st}^l_{X,Y \times Z}} & T(X \times Y \times Z) \\ & \searrow \mathbf{st}^l_{X,Y} \times 1 \quad \nearrow \mathbf{st}^l_{X \times Y, Z} & \\ & T(X \times Y) \times Z & \end{array}$$

and

$$\begin{array}{ccccc} X \times Y & & & & \\ \eta_X \times 1 \downarrow & \searrow \eta_{X \times Y} & & & \\ T(X) \times Y & \xrightarrow{\mathbf{st}^l_{X,Y}} & T(X \times Y) & & \\ \mu_X \times 1 \uparrow & & & \swarrow \mu_{X \times Y} & \\ T^2(X) \times Y & \xrightarrow{\mathbf{st}^l_{T(X),Y}} & T(T(X) \times Y) & \xrightarrow{T(\mathbf{st}^l_{X \times Y})} & T^2(X \times Y) \end{array}$$

commute.

The *right strength* is a natural transformation with components

$$\mathbf{st}^r_{X,Y} : X \times T(Y) \longrightarrow T(X \times Y)$$

for sets X and Y , making the diagrams

$$\begin{array}{ccc} X \times Y \times T(Z) & \xrightarrow{\mathbf{st}^r_{X \times Y, Z}} & T(X \times Y \times Z) \\ & \searrow 1 \times \mathbf{st}^r_{Y,Z} \quad \nearrow \mathbf{st}^r_{X, Y \times Z} & \\ & X \times T(Y \times Z) & \end{array}$$

and

$$\begin{array}{ccccc}
X \times Y & & & & \\
\downarrow 1 \times \eta_Y & \searrow \eta_{X \times Y} & & & \\
X \times T(Y) & \xrightarrow{\mathbf{st}^r_{X,Y}} & T(X \times Y) & & \\
\uparrow 1 \times \mu_Y & & & \swarrow \mu_{X \times Y} & \\
X \times T^2(Y) & \xrightarrow{\mathbf{st}^r_{X,T(Y)}} & T(X \times T(Y)) & \xrightarrow{T(\mathbf{st}^r_{X \times Y})} & T^2(X \times Y)
\end{array}$$

commute.

The monad (T, η, μ) on Set together with two natural transformations \mathbf{st}^l and \mathbf{st}^r of right and left strength is a *bi-strong monad* if, for any sets X, Y, Z , the square

$$\begin{array}{ccc}
X \times T(Y) \times Z & \xrightarrow{1_X \times \mathbf{st}^l_{Y,Z}} & X \times T((Y \times Z)) \\
\downarrow \mathbf{st}^r_{X,Y} \times 1_Z & & \downarrow \mathbf{st}^r_{X,Y \times Z} \\
T(X \times Y) \times Z & \xrightarrow{\mathbf{st}^l_{X \times Y, Z}} & T(X \times Y \times Z)
\end{array}$$

commutes.

As we already mentioned, each monad (T, η, μ) on Set is bi-strong. We shall define the right and left strength. Fix sets X and Y . For $x \in X$ and $y \in Y$, we have functions

$$l_y : X \longrightarrow X \times Y, \quad \text{and} \quad r_x : Y \longrightarrow X \times Y,$$

such that

$$l_y(x) = \langle x, y \rangle, \quad \text{and} \quad r_x(y) = \langle x, y \rangle.$$

The left and right strength

$$\mathbf{st}^l_{X,Y} : T(X) \times Y \longrightarrow T(X \times Y) \quad \text{and} \quad \mathbf{st}^r_{X,Y} : X \times T(Y) \longrightarrow T(X \times Y)$$

are given for $x \in X$, $s \in T(X)$, $y \in Y$ and $t \in T(Y)$ by

$$\mathbf{st}^l_{X,Y}(s, y) = T(l_y)(s) \quad \text{and} \quad \mathbf{st}^r_{X,Y}(x, t) = T(r_x)(t),$$

respectively. We drop indices $_{X,Y}$ when it does not lead to confusion.

It is not difficult to verify that the above defines left (\mathbf{st}^l) and right (\mathbf{st}^r) strength on the monad T and since, for any $x \in X$ and $z \in Z$, the square

$$\begin{array}{ccc}
Y & \xrightarrow{r_x} & X \times Y \\
l_z \downarrow & & \downarrow l_z \\
Y \times Z & \xrightarrow{r_x} & X \times Y \times Z
\end{array}$$

commutes, they are compatible and make the monad T bi-strong. Note that these strengths are related by the following diagram

$$\begin{array}{ccc}
T(X) \times Y & \xrightarrow{\mathbf{st}^l_{X,Y}} & T(X \times Y) \\
T(\langle \pi_2, \pi_1 \rangle) \downarrow & & \uparrow \langle \pi_2, \pi_1 \rangle \\
Y \times T(X) & \xrightarrow{\mathbf{st}^r_{Y,X}} & T(Y \times X)
\end{array}$$

Examples of strength on monads in Set.

1. Maybe monad. The left strength $\mathbf{st}^l_{X,Y} : (X + \{\perp\}) \times Y \longrightarrow (X \times Y) + \{\perp\}$ is given by

$$\mathbf{st}^l(x, y) = \begin{cases} \perp & \text{if } x = \perp \\ \langle x, y \rangle & \text{otherwise.} \end{cases}$$

Right strength is similar.

2. List monad. The left strength $\mathbf{st}^l : T(X) \times Y \longrightarrow T(X \times Y)$ is given by

$$\mathbf{st}^l(\vec{x}, y) = \begin{cases} \varepsilon & \text{if } \vec{x} = \varepsilon \\ \langle x_1, y \rangle, \dots, \langle x_n, y \rangle & \text{if } \vec{x} = x_1, \dots, x_n. \end{cases}$$

Right strength is similar.

3. Continuation monad. We shall describe the strength morphisms by lambda terms. The left strength is

$$\mathbf{st}^l = \lambda N_{:\mathcal{C}(X)} . \lambda y_{:Y} . \lambda c_{:\mathcal{P}(X \times Y)} . N(\lambda x_{:X} . c(x, y)) : \mathcal{C}(X) \times Y \longrightarrow \mathcal{C}(X \times Y)$$

and the right strength is

$$\mathbf{st}^r = \lambda x_{:X} . \lambda M_{:\mathcal{C}(Y)} . \lambda c_{:\mathcal{P}(X \times Y)} . M(\lambda y_{:Y} . c(x, y)) : X \times \mathcal{C}(Y) \longrightarrow \mathcal{C}(X \times Y).$$

2.5 Combining computations in arbitrary monad T on Set

Using both strengths, we can define two *pile up* natural transformations, the left and right. For any sets X and Y , the *left pile up* $\mathbf{pile'up}^l_{X,Y}$ is defined from the diagram

$$\begin{array}{ccc}
 T(X) \times T(Y) & \xrightarrow{\mathbf{pile'up}^l_{X,Y}} & T(X \times Y) \\
 \mathbf{st}^l_{X,T(Y)} \downarrow & & \uparrow \mu_{X \times Y} \\
 T(X \times T(Y)) & \xrightarrow{T(\mathbf{st}^r_{X,Y})} & T^2(X \times Y)
 \end{array}$$

In the above diagram, the function $\mathbf{pile'up}^l_{X,Y}$ is defined as a composition of three operations: the first is taking the T -computation on X ‘outside’ to be a computation on $X \times T(Y)$, the second is taking the T -computation on Y ‘outside’ to be a T -computation on $X \times Y$. In this way, we have T -computations coming from X on T -computations coming from Y on $X \times Y$. Now the last morphism $\mu_{X \times Y}$ flattens these two levels to one, i.e. the T -computation on T -computations to T -computations.

The *right pile up* $\mathbf{pile'up}^r_{X,Y}$ is defined from the diagram

$$\begin{array}{ccc}
 T(X) \times T(Y) & \xrightarrow{\mathbf{pile'up}^r_{X,Y}} & T(X \times Y) \\
 \mathbf{st}^r_{T(X),Y} \downarrow & & \uparrow \mu_{X \times Y} \\
 T(T(X) \times Y) & \xrightarrow{T(\mathbf{st}^l_{X,Y})} & T^2(X \times Y)
 \end{array}$$

This operation takes out the T -computations in a reverse order and so they pile up in the other way.

If these *pile up* operations agree for all sets X and Y , the monad is called *commutative*. On our list of monads, identity, maybe and covariant power-set monads are commutative. The exception, list and continuation monads are not commutative. Most monads, including the continuation monad \mathcal{C} , are not commutative. It should be noticed that even if the monad T is not commutative, both lift morphisms agree on pairs in which at least one component comes from the actual value (not an arbitrary T -computation). In other words, the functions

$$T(X_1) \times T(X_2) \xrightleftharpoons[\text{pile'up}^r_{X_1, X_2}]{\text{pile'up}^l_{X_1, X_2}} T(X_1 \times X_2)$$

are equalized by both

$$X_1 \times T(X_2) \xrightarrow{\eta_{X_1} \times 1} T(X_1) \times T(X_2)$$

and

$$T(X_1) \times X_2 \xrightarrow{1 \times \eta_{X_2}} T(X_1) \times T(X_2)$$

morphisms. Both pile'up^l and pile'up^r are associative. All this is shown in the Appendix.

Examples of pile'up -operations.

1. Maybe monad. The left and right pile'up 's coincide in this case, as in any commutative monad. We have $\text{pile'up}^l_{X,Y} = \text{pile'up}^r_{X,Y} : (X + \{\perp\}) \times (Y + \{\perp'\}) \longrightarrow (X \times Y) + \{\perp\}$ is given by

$$\text{st}^l(x, y) = \begin{cases} \perp & \text{if } \{x, y\} \cap \{\perp, \perp'\} \neq \emptyset \\ \langle x, y \rangle & \text{otherwise.} \end{cases}$$

2. List monad. The left $\text{pile'up}^l : T(X) \times T(Y) \longrightarrow T(X \times Y)$ is given by

$$\begin{aligned} \text{pile'up}^l(\langle x_1, \dots, x_n \rangle, \langle y_1, \dots, y_m \rangle) &= \\ &= \langle \langle x_1, y_1 \rangle, \langle x_1, y_2 \rangle, \dots, \langle x_1, y_m \rangle, \langle x_2, y_1 \rangle, \dots, \langle x_n, y_{m-1} \rangle, \langle x_n, y_m \rangle \end{aligned}$$

and the right $\text{pile'up}^r : T(X) \times T(Y) \longrightarrow T(X \times Y)$ is given by

$$\begin{aligned} \text{pile'up}^r(\langle x_1, \dots, x_n \rangle, \langle y_1, \dots, y_m \rangle) &= \\ &= \langle \langle x_1, y_1 \rangle, \langle x_2, y_1 \rangle, \dots, \langle x_n, y_1 \rangle, \langle x_1, y_2 \rangle, \dots, \langle x_{n-1}, y_m \rangle, \langle x_n, y_m \rangle. \end{aligned}$$

3. (Covariant) power-set monad. The left and right pile'up 's coincide in this case. We have $\text{pile'up}^l_{X,Y} = \text{pile'up}^r_{X,Y} : \mathcal{P}(X) \times \mathcal{P}(Y) \longrightarrow \mathcal{P}(X \times Y)$ given by

$$\text{st}^l(U, V) = U \times V$$

for $U \in \mathcal{P}(X)$ and $V \in \mathcal{P}(Y)$.

4. Continuation monad. Both **pile'up** operations

$$\mathbf{pile'up}^l, \mathbf{pile'up}^r : \mathcal{C}(X) \times \mathcal{C}(Y) \longrightarrow \mathcal{C}(X \times Y)$$

can be defined, for $M \in \mathcal{C}(X)$ and $N \in \mathcal{C}(Y)$, by lambda terms as

$$\mathbf{pile'up}^l(M, N) = \lambda c : \mathcal{P}(X \times Y). M(\lambda x : X. N(\lambda y : Y. c(x, y)))$$

and

$$\mathbf{pile'up}^r(M, N) = \lambda c : \mathcal{P}(X \times Y). N(\lambda y : Y. M(\lambda x : X. c(x, y))).$$

The calculations for these operations are in the Appendix.

Thus in the case of the continuation monad ‘piling up’ computations one on top of the other is nothing but putting (interpretations of) quantifiers (= computations in the continuation monad) in order, either first before the second or the second before the first.

2.6 T -transforms on arbitrary monad T on Set

There are two (binary) T -transformations, right and left. For a function $f : X \times Y \longrightarrow T(Z)$, the left T -transform is defined as the composition

$$\begin{array}{ccc} T(X) \times T(Y) & \xrightarrow{\mathbf{TR}^{l,T}_{X,Y}(f)} & T(Z) \\ \mathbf{pile'up}^l \downarrow & & \uparrow \mu_Z \\ T(X \times Y) & \xrightarrow{T(f)} & T^2(Z) \end{array}$$

and the right T -transform is defined as the composition

$$\begin{array}{ccc} T(X) \times T(Y) & \xrightarrow{\mathbf{TR}^{r,T}_{X,Y}(f)} & T(Z) \\ \mathbf{pile'up}^r \downarrow & & \uparrow \mu_Z \\ T(X \times Y) & \xrightarrow{T(f)} & T^2(Z) \end{array}$$

The most popular **CPS**-transforms are for the evaluation morphism $ev : X \times (X \Rightarrow Y) \rightarrow Y$ but there are also other morphisms having useful transforms.

*Examples of T -transforms and in particular **CPS**-transforms.*

1. The evaluation map $ev : X \times (X \Rightarrow Y) \rightarrow Y$ gives rise to application transforms

$$\mathbf{TR}^{l,T}(ev), \mathbf{TR}^{r,T}(ev) : T(X) \times T(X \Rightarrow Y) \rightarrow T(Y).$$

In case T is the continuation monad \mathcal{C} , they are the usual **CPS**-transforms $\mathbf{CPS}^l(ev), \mathbf{CPS}^r(ev) : \mathcal{C}(X) \times \mathcal{C}(X \Rightarrow Y) \rightarrow \mathcal{C}(Y)$ given by

$$\mathbf{CPS}^l(ev)(M, N) = \lambda h_{:\mathcal{P}(Y)}. M(\lambda x_{:X}. N(\lambda g_{:X \Rightarrow Y}. h(g\ x)))$$

for $M \in \mathcal{C}(X)$ and $N \in \mathcal{C}(X \Rightarrow Y)$.

Right version is similar.

2. Various epsilon maps are typically defined as maps from a product. Thus they give rise to various T -transforms. We list some of them below mainly to introduce notation that will be used later. The definitions are given by lambda terms.

- (a) Left evaluation

$$\mathbf{eps}^l_X = \lambda h_{:\mathcal{P}(X)}. \lambda x_{:X}. h(x) : \mathcal{P}(X) \times X \rightarrow \mathbf{t};$$

- (b) Right evaluation

$$\mathbf{eps}^r_X = \lambda x_{:X}. \lambda h_{:\mathcal{P}(X)}. h(x) : X \times \mathcal{P}(X) \rightarrow \mathbf{t};$$

- (c) Left partial evaluation

$$\mathbf{eps}^{l,X}_Y = \mathbf{eps}^l_Y = \lambda c_{:\mathcal{P}(X \times Y)}. \lambda y_{:Y}. \lambda x_{:X}. c(x, y) : \mathcal{P}(X \times Y) \times Y \rightarrow \mathcal{P}(X);$$

- (d) Right partial evaluation

$$\mathbf{eps}^{r,X}_Y = \mathbf{eps}^r_Y = \lambda y_{:Y}. \lambda c_{:\mathcal{P}(X \times Y)}. \lambda x_{:X}. c(x, y) : Y \times \mathcal{P}(X \times Y) \rightarrow \mathcal{P}(X);$$

3. What we call Mostowski maps are maps similar to **eps**'es that are the algebraic counterpart of the interpretation of generalized quantifiers of Mostowski. Again, we give a definition for total and partial case.

- (a) Left Mostowski

$$\mathbf{mos}^l_X = \lambda Q_{:\mathcal{C}(X)}. \lambda c_{:\mathcal{P}(X)}. Q(c) : \mathcal{C}(X) \times \mathcal{P}(X) \rightarrow \mathbf{t};$$

(b) Right Mostowski

$$\mathbf{mos}^r_X = \lambda c:\mathcal{P}(X).\lambda Q:\mathcal{C}(X).Q(c) : \mathcal{P}(X) \times \mathcal{C}(X) \rightarrow \mathbf{t};$$

(c) Left partial Mostowski

$$\mathbf{mos}^l_Y = \lambda Q:\mathcal{C}(Y).\lambda c:\mathcal{P}(X \times Y).\lambda x:X.Q(\lambda y:Y.c(x, y)) : \mathcal{C}(Y) \times \mathcal{P}(X \times Y) \rightarrow \mathcal{P}(X);$$

(d) Right partial Mostowski

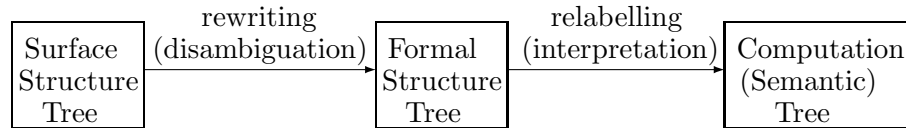
$$\mathbf{mos}^r_Y = \lambda c:\mathcal{P}(X \times Y).\lambda Q:\mathcal{C}(Y).\lambda x:X.Q(\lambda y:Y.c(x, y)) : \mathcal{P}(X \times Y) \times \mathcal{C}(Y) \rightarrow \mathcal{P}(X).$$

3 Scope assignment strategies

Using the notions connected to the continuation monad introduced above, we shall now precisely state and compare three strategies (A, B, C) for determining the meaning of multi-quantifier sentences.

3.1 General remarks

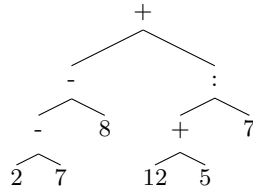
In each strategy, the starting point is the Surface Structure Tree of a sentence. This tree is rewritten so as to obtain Formal Structure Trees that correspond to all and only the available meanings of the sentence. Finally, we relabel those trees to obtain Computation Trees¹ that provide the semantics for the sentence in each of its reading.



¹We think of Computation Trees by analogy with mathematical expressions, e.g.

$$((2 - 7) - 8) + ((12 + 5) : 7)$$

that can be represented as

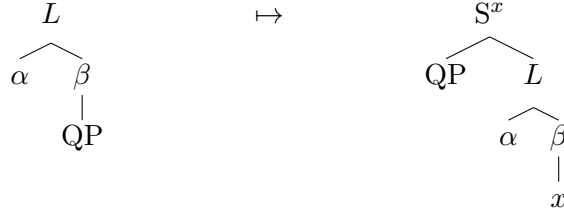


i.e. a labeled binary tree where the leaves of this tree are labeled with values and the internal nodes are labeled with operations that will be applied in the computation to the values obtained from the computations of the left and right subtrees.

Rewriting. Scope assignment strategies can be divided into two families: movement analyses (rewriting rules include QR, Predicate Collapsing and possibly Rotation) and in situ analyses (no rewriting rules). Below we define three rewrite rules on trees: QR Rule, Predicate Collapsing and Rotation.

- QR (Quantifier Raising) Rule

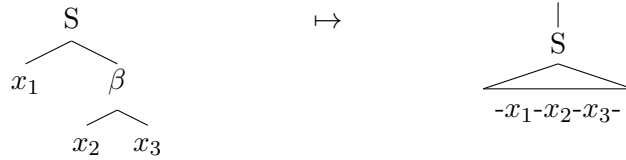
- applies when we have a chosen QP in a leaf of a tree;
- adjoins QP to S;
- indexes S with the variable bound by the raised QP.



(L - label, α, β - subtrees.)

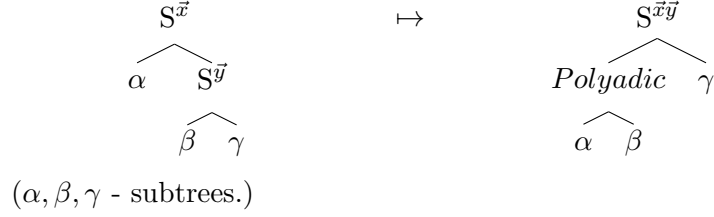
- Predicate Collapsing

- applies when all the leaves under the node labeled S are labeled with variables (not QP's);
- collapses the whole subtree with the root S to a single leaf labeled with the variables x_1, x_2, x_3 from the leaves under S-node.



- Rotation

- applies to a tree with two distinguished nodes labeled with S's superscripted with some variables: the mother labeled $S^{\vec{x}}$ and its right daughter labeled $S^{\vec{y}}$;
- it rotates left the subtree with root labeled $S^{\vec{x}}$;
- the root of this subtree is labeled $S^{\vec{x}\vec{y}}$ and the (new) left daughter is labeled *Polyadic*.



Relabelling. In each scope assignment strategy, the leaves in the Computation Tree have the same labels: QPs are interpreted as \mathcal{C} -computations, and predicates are interpreted as usual or lifted. The main difference among the three approaches consists in the shape of the Formal Structure Trees and the operations (**eps**'es, **mos**'es, **pile'up**'es, **CPS**'es) used as labels of the inner nodes of the Computation Trees.

3.2 Strategy A

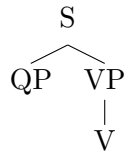
In the traditional movement strategy (as variously implemented in [16], [19])

- Surface Structure Tree gets rewritten (disambiguated) as Formal Structure Trees (Logical Forms) via
 - QR Rule;
 - Predicate Collapsing.
- Formal Structure Trees (LFs) are relabelled as Computation Trees as follows
 - S^x (root of a subtree representing a formula) is interpreted as a suitably typed **mos**-operation (the only operation allowed);
 - S (leaf of a tree) is interpreted as a predicate;
 - QP (leaf of a tree) is interpreted as a generalized quantifier $\|Q\|$ quantifying over a set X (i.e. as a \mathcal{C} -computation on X).

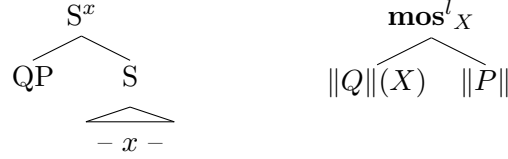
We will illustrate each strategy on the examples involving one, two and three QPs.

Sentence with one QP, e.g. *Every kid (most kids) entered.*

(A1) Surface Structure Tree



(A1) Formal Structure Tree (LF) and the corresponding Computation Tree



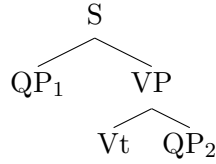
The Computation Tree in (A1) gives rise to the following general map

$$\begin{array}{ccc} & \mathcal{C}(X) \times \mathcal{P}(X) & \\ \mathbf{strat}_A^1 : & \downarrow \mathbf{mos}^l_X & \\ & 2 & \end{array}$$

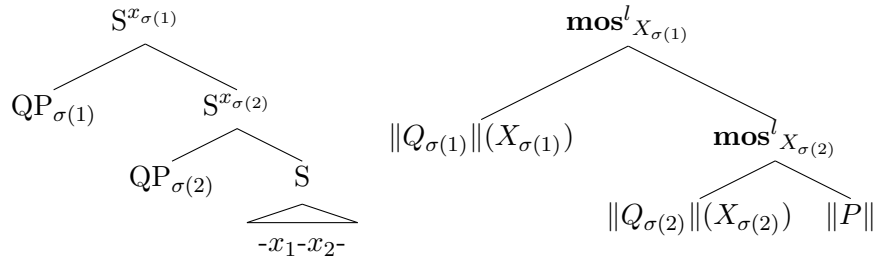
In this case, there is one such map - thus Strategy A yields one reading for a sentence with one QP.

Sentence with two QPs, e.g. *Every girl likes a boy.*

(A2) Surface Structure Tree



(A2) Formal Structure Tree (LF) and the corresponding Computation Tree



The Computation Tree in (A2) gives rise to the following general map, with $\sigma \in S_2$ (where S_2 is the set of permutations of the set $\{1, 2\}$)

$$\begin{array}{c}
\mathcal{C}(X_1) \times \mathcal{P}(X_1 \times X_2) \times \mathcal{C}(X_2) \xrightarrow{\bar{\pi}_{\sigma(i)}} \mathcal{C}(X_{\sigma(i)}) \\
\downarrow \langle \bar{\pi}_{\sigma(1)}, \bar{\pi}_{\sigma(2)}, \pi_2 \rangle \\
\mathcal{C}(X_{\sigma(1)}) \times \mathcal{C}(X_{\sigma(2)}) \times \mathcal{P}(X_1 \times X_2) \\
\downarrow 1 \times \mathbf{mos}^l_{X_{\sigma(2)}} \\
\mathcal{C}(X_{\sigma(1)}) \times \mathcal{P}(X_{\sigma(1)}) \\
\downarrow \mathbf{mos}^l_{X_{\sigma(1)}} \\
2
\end{array}$$

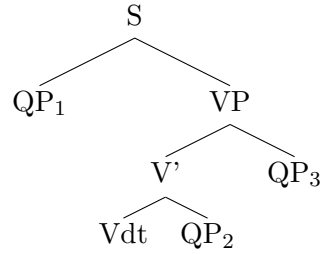
strat_A^{2,σ} :

where $\bar{\pi}_{\sigma(i)}$ is the projection on the 1st factor if $\sigma(i) = 1$ and on the 3rd factor if $\sigma(i) = 2$, i.e. as it should be. This convention will be used in all similar diagrams without any further explanations.

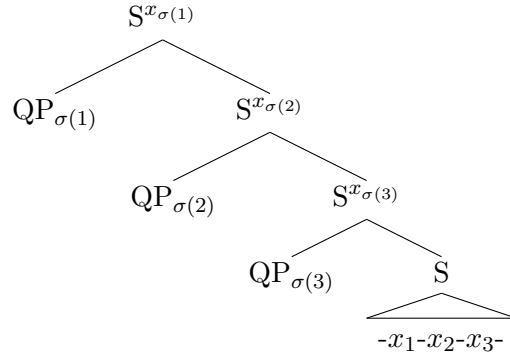
There are two such maps corresponding to the two permutations σ of $\{1, 2\}$. These maps are different in general. Thus Strategy A yields two (both) asymmetric readings for a sentence with two QPs.

Sentence with three QPs, e.g. *Some teacher gave every student most books.*

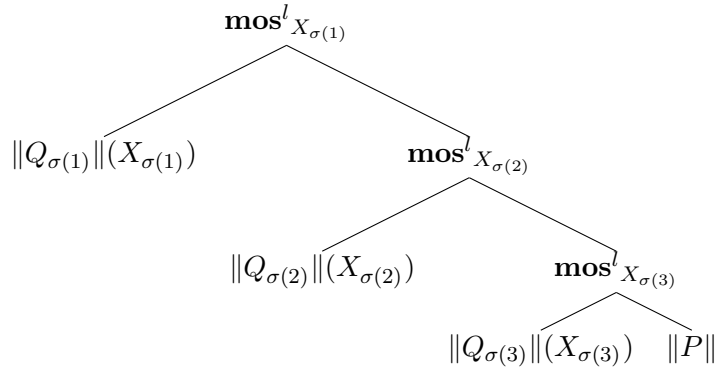
(A3) Surface Structure Tree



(A3) Formal Structure Tree (LF)



and the corresponding Computation Tree



The Computation Tree in (A3) gives rise to the following general map, with $\sigma \in S_3$ (where S_3 is the set of permutations of the set $\{1, 2, 3\}$)

$$\begin{array}{c}
\mathcal{C}(X_1) \times \mathcal{P}(X_1 \times X_2 \times X_3) \times \mathcal{C}(X_2) \times \mathcal{C}(X_3) \\
\downarrow \langle \bar{\pi}_{\sigma(1)}, \bar{\pi}_{\sigma(2)}, \bar{\pi}_{\sigma(3)}, \pi_2 \rangle \\
\mathcal{C}(X_{\sigma(1)}) \times \mathcal{C}(X_{\sigma(2)}) \times \mathcal{C}(X_{\sigma(3)}) \times \mathcal{P}(X_1 \times X_2 \times X_3) \\
\downarrow 1 \times 1 \times \mathbf{mos}^l_{X_{\sigma(3)}} \\
\mathcal{C}(X_{\sigma(1)}) \times \mathcal{C}(X_{\sigma(2)}) \times \mathcal{P}(\dots \times \widehat{X_{\sigma(3)}} \times \dots) \\
\downarrow 1 \times \mathbf{mos}^l_{X_{\sigma(2)}} \\
\mathcal{C}(X_{\sigma(1)}) \times \mathcal{P}(X_{\sigma(1)}) \\
\downarrow \mathbf{mos}^l_{X_{\sigma(1)}} \\
2
\end{array}$$

There are six such maps corresponding to six permutations σ of $\{1, 2, 3\}$. These maps are different in general. Thus Strategy A yields 6 asymmetric readings for a sentence with three QPs.

3.3 Strategy B

In the polyadic approach (as developed in [17], [9], [10], [22], [5])

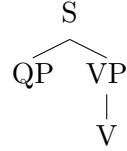
- Surface Structure Tree gets rewritten (disambiguated) as Formal Structure Trees (Polyadic Logical Forms) via
 - QR Rule;
 - Predicate Collapsing;
 - Rotation.
- Formal Structure Trees (PLFs) are relabelled as Computation Trees as follows
 - *Polyadic* (root of a subtree representing a polyadic quantifier) is interpreted as a suitably typed **pile'**up-operation (we can choose

globally whether we use only **pile'up**^{*l*} or **pile'up**^{*r*} and then consequently stick to it).

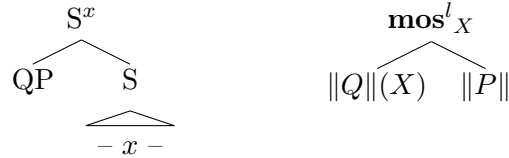
– S^x , S, QP are interpreted as above.

Sentence with one QP, e.g. *Every kid (most kids) entered.*

(B1) Surface Structure Tree



(B1) Formal Structure Tree (PLF) and the corresponding Computation Tree



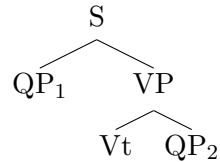
The Computation Tree in (B1) gives rise to the following general map

$$\begin{array}{ccc} & \mathcal{C}(X) \times \mathcal{P}(X) & \\ & \downarrow \mathbf{mos}^l_X & \\ \mathbf{strat}_B^1 : & & 2 \end{array}$$

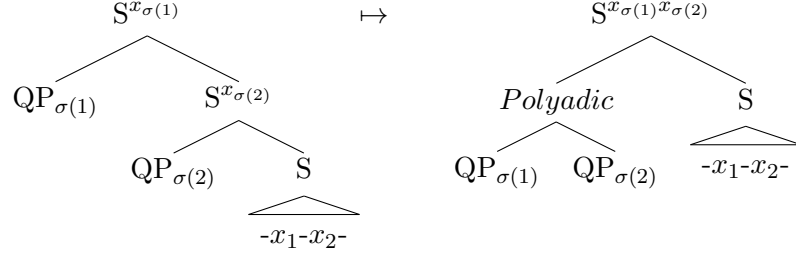
In this case, there is one such map - thus Strategy B yields one reading for a sentence with one QP.

Sentence with two QPs, e.g. *Every girl likes a boy.*

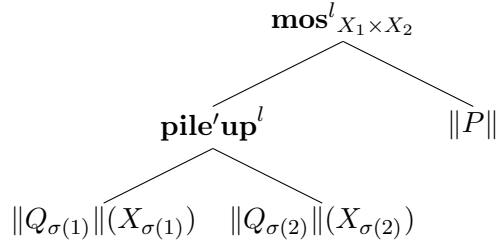
(B2) Surface Structure Tree



(B2) Formal Structure Tree (PLF) obtained from LF in (A2) via rotation



and the corresponding Computation Tree



The Computation Tree in (B2) gives rise to the following general map, with $\sigma \in S_2$

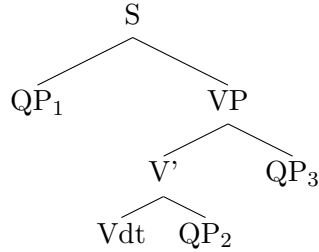
$$\begin{array}{c}
\mathcal{C}(X_1) \times \mathcal{P}(X_1 \times X_2) \times \mathcal{C}(X_2) \\
\downarrow \langle \bar{\pi}_{\sigma(1)}, \bar{\pi}_{\sigma(2)}, \pi_2 \rangle \\
\mathcal{C}(X_{\sigma(1)}) \times \mathcal{C}(X_{\sigma(2)}) \times \mathcal{P}(X_1 \times X_2) \\
\downarrow \mathbf{pile'up}^l \times 1 \\
\mathcal{C}(X_{\sigma(1)} \times X_{\sigma(2)}) \times \mathcal{P}(X_1 \times X_2) \\
\downarrow \mathcal{C}(\pi_{\sigma^{-1}}) \times 1 \\
\mathcal{C}(X_1 \times X_2) \times \mathcal{P}(X_1 \times X_2) \\
\downarrow \mathbf{mos}^l_{X_1 \times X_2} \\
2
\end{array}$$

$\mathbf{strat}_B^{2,\sigma} :$

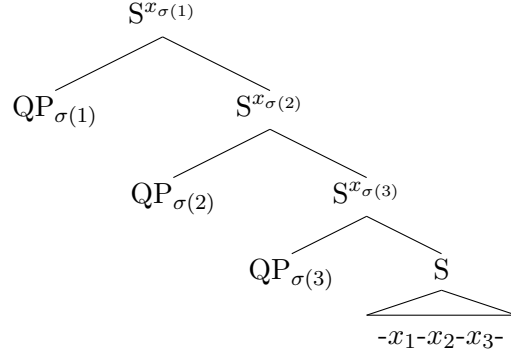
There are two such maps corresponding to the two permutations σ of $\{1, 2\}$ combined with $\mathbf{pile'up}^l$ -operation (in that case, alternatively, we can use $\mathbf{pile'up}^l$ and $\mathbf{pile'up}^r$). These maps are different in general. Thus Strategy B yields two (both) asymmetric readings for a sentence with two QPs.

Sentence with three QPs, e.g. *Some teacher gave every student most books.*

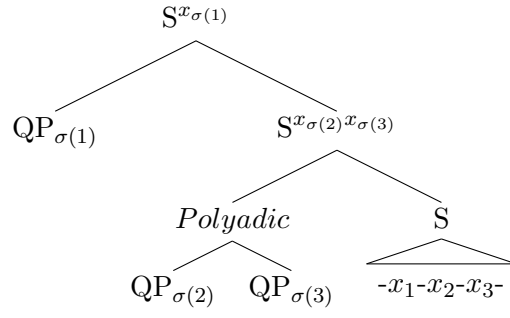
(B3) Surface Structure Tree



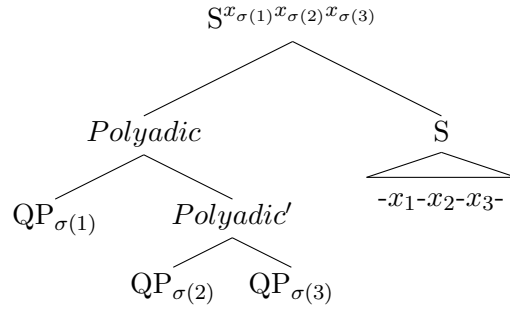
(B3) Formal Structure Tree (PLF) obtained from LF in (A3) via rotation



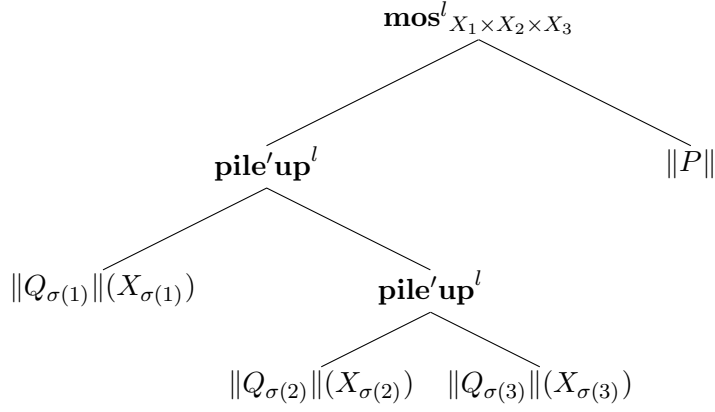
\mapsto



\mapsto



and the corresponding Computation Tree



The Computation Tree in (B3) gives rise to the following general map, with $\sigma \in S_3$

$$\begin{array}{c}
 \mathcal{C}(X_1) \times \mathcal{P}(X_1 \times X_2 \times X_3) \times \mathcal{C}(X_2) \times \mathcal{C}(X_3) \\
 \downarrow \langle \bar{\pi}_{\sigma(1)}, \bar{\pi}_{\sigma(2)}, \bar{\pi}_{\sigma(3)}, \pi_2 \rangle \\
 \mathcal{C}(X_{\sigma(1)}) \times \mathcal{C}(X_{\sigma(2)}) \times \mathcal{C}(X_{\sigma(3)}) \times \mathcal{P}(X_1 \times X_2 \times X_3) \\
 \downarrow 1 \times \mathbf{pile'up}^l \times 1 \\
 \mathcal{C}(X_{\sigma(1)}) \times \mathcal{C}(X_{\sigma(2)} \times X_{\sigma(3)}) \times \mathcal{P}(X_1 \times X_2 \times X_3) \\
 \downarrow \mathbf{pile'up}^l \times 1 \\
 \mathcal{C}(X_{\sigma(1)} \times X_{\sigma(2)} \times X_{\sigma(3)}) \times \mathcal{P}(X_1 \times X_2 \times X_3) \\
 \downarrow \mathcal{C}(\pi_{\sigma^{-1}}) \times 1 \\
 \mathcal{C}(X_1 \times X_2 \times X_3) \times \mathcal{P}(X_1 \times X_2 \times X_3) \\
 \downarrow \mathbf{mos}^l_{X_1 \times X_2 \times X_3} \\
 2
 \end{array}$$

There are six such maps corresponding to six permutations σ of $\{1, 2, 3\}$ combined with **pile'up**^{*l*}-operation (in that case we can choose globally whether we use only **pile'up**^{*l*} or **pile'up**^{*r*} and then consequently stick to it). These maps are different in general. Thus Strategy B yields 6 asymmetric readings for a sentence with three QPs.

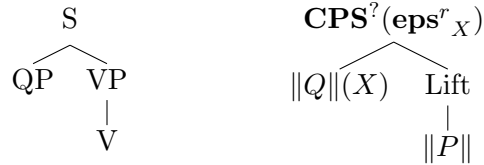
3.4 Strategy C

In the continuation-based strategy approach (as proposed in [1])

- Surface Structure Tree gets rewritten as Formal Structure Tree via
 - no rewriting rules (Formal Structure Trees are just Surface Structure Trees - this is what is understood by in situ).
- Relabelling Formal Structure Trees (= Surface Structure Trees) as the Computation Trees is as follows
 - S, VP, V' (roots of a (sub)tree with some (possibly all) arguments provided) are interpreted as suitably typed **CPS**-operations (left and right);
 - V, Vt, Vdt (leaves of a tree) are interpreted as ‘continuized’ (1-, 2-, 3-*ary*, respectively) predicates.

Sentence with one QP, e.g. *Every kid (most kids) entered.*

(C1) Surface Structure Tree and the corresponding Computation Tree



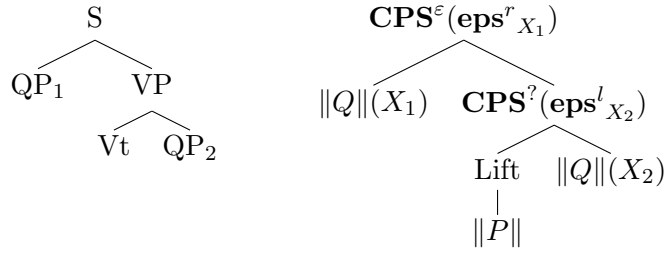
The Computation Tree in (C1) gives rise to the following general map

$$\begin{array}{ccc}
& \mathcal{C}(X) \times \mathcal{P}(X) & \\
\text{strat}_c^1 : & \downarrow 1 \times \eta_{\mathcal{P}(X)} & \\
& \mathcal{C}(X) \times \mathcal{CP}(X) & \\
& \downarrow \mathbf{CPS}^?(\mathbf{eps}^r_X) & \\
& \mathcal{C}(2) \xrightarrow{ev_{id_2}} 2 &
\end{array}$$

We use $\mathbf{CPS}^?$ when it does not matter whether we apply \mathbf{CPS}^l or \mathbf{CPS}^r . This is the case when one of the arguments is a lifted element (like interpretations of predicates in this strategy). Strategy C yields one reading for a sentence with one QPs.

Sentence with two QPs, e.g. *Every girl likes a boy*.

(C2) Surface Structure Tree and the corresponding Computation Tree



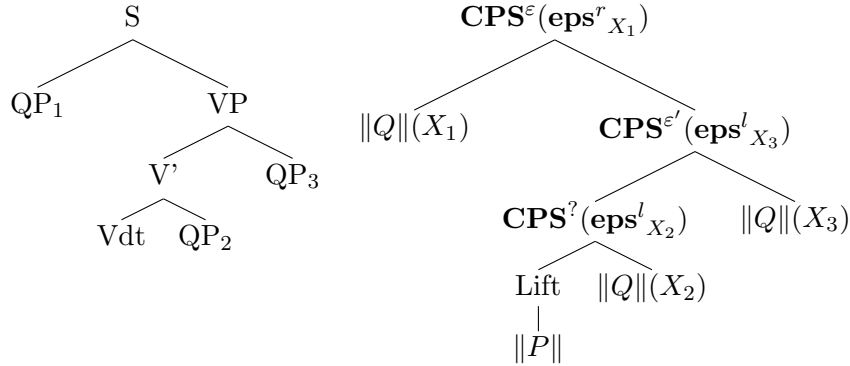
The Computation Tree in (C2) gives rise to the following general map

$$\begin{array}{c}
\mathcal{C}(X_1) \times \mathcal{P}(X_1 \times X_2) \times \mathcal{C}(X_2) \\
\downarrow 1 \times \eta_{\mathcal{P}(X_1 \times X_2)} \times 1 \\
\mathcal{C}(X_1) \times \mathcal{CP}(X_1 \times X_2) \times \mathcal{C}(X_2) \\
\downarrow 1 \times \mathbf{CPS}^?(\mathbf{eps}^l_{X_2}) \\
\mathcal{C}(X_1) \times \mathcal{CP}(X_1) \\
\downarrow \mathbf{CPS}^\varepsilon(\mathbf{eps}^r_{X_1}) \\
\mathcal{C}(2) \xrightarrow{ev_{id_2}} 2
\end{array}$$

with $\varepsilon \in \{l, r\}$. Depending on whether we use \mathbf{CPS}^l or \mathbf{CPS}^r , we get either one or the other of the two asymmetric readings for a sentence with two QPs. Strategy C yields two readings for a sentence with two QPs corresponding to the two \mathbf{CPS} 'es.

Sentence with three QPs, e.g. *Some teacher gave every student most books.*

(C3) Surface Structure Tree and the corresponding Computation Tree



The Computation Tree in (C3) gives rise to the following general map

$$\begin{array}{c}
\mathcal{C}(X_1) \times \mathcal{P}(X_1 \times X_2 \times X_3) \times \mathcal{C}(X_2) \times \mathcal{C}(X_3) \\
\downarrow 1 \times \eta_{\mathcal{P}(X_1 \times X_2 \times X_3)} \times 1 \times 1 \\
\mathcal{C}(X_1) \times \mathcal{CP}(X_1 \times X_2 \times X_3) \times \mathcal{C}(X_2) \times \mathcal{C}(X_3) \\
\downarrow 1 \times \mathbf{CPS}^?(\mathbf{eps}^l_{X_2}) \times 1 \\
\mathcal{C}(X_1) \times \mathcal{CP}(X_1 \times X_3) \times \mathcal{C}(X_3) \\
\downarrow 1 \times \mathbf{CPS}^{\varepsilon'}(\mathbf{eps}^l_{X_3}) \\
\mathcal{C}(X_1) \times \mathcal{CP}(X_1) \\
\downarrow \mathbf{CPS}^{\varepsilon}(\mathbf{eps}^r_{X_1}) \\
\mathcal{C}(2) \xrightarrow{ev_{id_2}} 2
\end{array}$$

$\mathbf{strat}_C^{3, \varepsilon', \varepsilon} :$

Strategy C provides four asymmetric readings for the sentence such that QP in subject position can be placed either first or last only (corresponding to the four possible combinations of the two **CPS**'es). Thus it yields four out of six readings accounted for by strategies A and B.

The tables below summarize the main features of the three approaches.

Passing from Surface Structure Tree Trees to Formal Structure Trees

<i>Strategy</i>	<i>A</i>	<i>B</i>	<i>C</i>
<i>Rewrite rules</i>	<i>QR, Predicate Collapsing</i>	<i>QR, Predicate Collapsing, Rotation</i>	<i>No rewrite rules (in situ)</i>

Passing from Formal Structure Trees to Computation Trees

<i>Strategy</i>	<i>A</i>	<i>B</i>	<i>C</i>
<i>Relabelling inner nodes</i>	$S^x \mapsto \mathbf{mos}$	$S^{\vec{x}} \mapsto \mathbf{mos}$ $Polyadic \mapsto$ pile'up	$S, VP, V' \mapsto$ CPS
<i>Relabelling leaves</i>	$S \mapsto relation$ $QP \mapsto C\text{-}comp.$	$S \mapsto relation$ $QP \mapsto C\text{-}comp.$	$V, Vt, Vdt \mapsto$ <i>continuized relation</i> $QP \mapsto C\text{-}comp.$

The semantics for sentences with intransitive and transitive verbs, as defined by the strategies A, B, and C, are equivalent. The semantics for sentences with ditransitive verbs, as defined by the strategies A, B, are equivalent. They provide six asymmetric readings of the sentence. The semantics for sentences with ditransitive verbs, as defined by the strategy C, provides four asymmetric readings of the sentence such that QP in subject position can be placed either first or last only. Thus they correspond to four out of six readings accounted for by strategies A and B. The proofs are given in the Appendix.

4 Appendix

4.1 The continuation monad

In this subsection, we gather all the basic facts (sometimes repeated from the text) of the *continuation monad* \mathcal{C} on Set . We have an adjunction

$$Set \begin{array}{c} \xrightarrow{\mathcal{P}} \\ \xleftarrow{\mathcal{P}^{op}} \end{array} Set^{op}$$

where both \mathcal{P} and \mathcal{P}^{op} are the contravariant powerset functors² with the domains and codomains as displayed. In particular, for $f : X \rightarrow Y$, the function $\mathcal{P}(f) = f^{-1} : \mathcal{P}(Y) \rightarrow \mathcal{P}(X)$ is given by

$$f^{-1}(h) = h \circ f$$

²Note that this is in contrast with the functor \mathcal{P} , where \mathcal{P} is the covariant power-set functor.

for $h : Y \rightarrow \mathbf{t}$.

Function $\eta_X : X \rightarrow \mathcal{C}(X)$, the component at set X of the unit of this adjunction $\eta : 1_{Set} \rightarrow \mathcal{P}\mathcal{P}^{op} = \mathcal{C}$, is given by

$$\eta_X(x) = \lambda h_{:\mathcal{P}(X)}.h(x).$$

Function $\varepsilon_X : X \rightarrow \mathcal{C}(X)$, the component at set X of the counit of this adjunction $\varepsilon : 1_{Set} \rightarrow \mathcal{P}^{op}\mathcal{P}$, is given by (essentially the same formula)

$$\varepsilon_X(x) = \lambda h_{:\mathcal{P}^{op}(X)}.h(x)$$

for $x \in X$.

The function $\mathcal{C}(f) : \mathcal{C}(X) \rightarrow \mathcal{C}(Y)$, for $Q : \mathcal{P}(X) \rightarrow \mathbf{t} \in \mathcal{C}(X)$, is a function $\mathcal{C}(f)(Q) : \mathcal{P}(Y) \rightarrow \mathbf{t}$ given by

$$\mathcal{C}(f)(Q)(h) = Q(h \circ f)$$

for $h : Y \rightarrow \mathbf{t}$.

The monad induced by this adjunction is the continuation monad. Its multiplication is given by the counit of the above adjunction transported back to *Set*, i.e. $\mu = \mathcal{P}^{op}(\varepsilon_{\mathcal{P}})$. For X in *Set*, the function

$$\mu_X : \mathcal{C}^2(X) \rightarrow \mathcal{C}(X)$$

is given by

$$\mu_X(\mathcal{R}) = \mathcal{R} \circ \eta_{\mathcal{P}(X)}$$

for $\mathcal{R} \in \mathcal{C}^2(X)$.

In λ -notation we write

$$\mu_X(\mathcal{F})(h) = \mathcal{F}(\lambda D_{:\mathcal{C}(X)}.D(h)).$$

The left strength for the monad \mathcal{C} is

$$\mathbf{st}^l : \mathcal{C}(X) \times Y \longrightarrow \mathcal{C}(X \times Y)$$

for $M \in \mathcal{C}(X)$ and $y \in Y$, given by

$$\mathbf{st}^l(M, y) = \lambda c_{:\mathcal{P}(X \times Y)}.M(\lambda x_{:X}.c(x, y)) : \mathcal{P}(X \times Y) \rightarrow \mathbf{t}$$

and the right strength, for $x \in X$ and $n \in \mathcal{C}(Y)$, is given by

$$\mathbf{st}^r(x, N) = \lambda c_{:\mathcal{P}(X \times Y)}.M(\lambda y_{:Y}.c(x, y)) : \mathcal{P}(X \times Y) \rightarrow \mathbf{t}.$$

The left pile'up operation

$$\mathbf{pile'up}^l : \mathcal{C}(X) \times \mathcal{C}(Y) \longrightarrow \mathcal{C}(X \times Y)$$

is the following composition

$$\mathcal{C}(X) \times \mathcal{C}(Y) \xrightarrow{\mathbf{st}^l} \mathcal{C}(X \times \mathcal{C}(Y)) \xrightarrow{\mathcal{C}(\mathbf{st}^r)} \mathcal{C}^2(X \times Y) \xrightarrow{\mu_{X \times Y}} \mathcal{C}(X \times Y)$$

where, for $Q \in \mathcal{C}(X)$, $Q' \in \mathcal{C}(Y)$, $c \in \mathcal{P}(X \times \mathcal{C}(Y))$, we have

$$\mathbf{st}^l(Q, Q')(c) = Q(\lambda x.X c(x, Q))$$

and, for $d \in \mathcal{C}(X \times \mathcal{C}(Y))$, $\mathcal{U} \in \mathcal{PC}(X \times Y)$, we have

$$\mathcal{C}(\mathbf{st}^r)(d)(\mathcal{U}) = d(\mathcal{U} \circ \mathbf{st}^r).$$

Now, using the above formulas, we can calculate $\mathbf{pile'up}^l$ as the composition on $Q \in \mathcal{C}(X)$, $Q' \in \mathcal{C}(Y)$, and $c \in \mathcal{P}(X \times Y)$ as follows

$$\begin{aligned} \mathbf{pile'up}^l(Q, Q')(c) &= \\ &= \mu_{X \times Y}(\mathcal{C}(\mathbf{st}^r)(\mathbf{st}^l(Q, Q')))(c) = \\ &= \mathcal{C}(\mathbf{st}^r)(\mathbf{st}^l(Q, Q'))(\lambda D:\mathcal{C}(X \times Y) D(c)) = \\ &= \mathbf{st}^l(Q, Q')((\lambda D:\mathcal{C}(X \times Y) D(c)) \circ \mathbf{st}^r) = \\ &= Q(\lambda x:X ((\lambda D:\mathcal{C}(X \times Y) D(c)) \circ \mathbf{st}^r)(x, Q')) = \\ &= Q(\lambda x:X ((\lambda D:\mathcal{C}(X \times Y) D(c))(\mathbf{st}^r(x, Q')))) = \\ &= Q(\lambda x.X \mathbf{st}^r(x, Q')(c)) = \\ &= Q(\lambda x.X Q'(\lambda y.Y c(x, y))) \end{aligned}$$

Similarly, we can show that

$$\mathbf{pile'up}^r(Q, Q')(c) = Q'(\lambda y.Y \mathcal{Q}(\lambda x.X c(x, y))).$$

One can easily verify that $\mathbf{pile'ups}$ are related by

$$\mathbf{pile'up}^r_{X,Y} = \mathcal{C}(\pi_{(2,1)}) \circ \mathbf{pile'up}^l_{Y,X} \circ \pi_{(2,1)}.$$

4.2 Some properties of $\mathbf{pile'up}$ operations

Lemma 4.1 (Pile'up lemma) *$\mathbf{pile'up}$'s on pairs where one element is continuaized agree and are equal to the corresponding strength.*

Proof. We have to show that the functions

$$T(X_1) \times T(X_2) \xrightleftharpoons[\text{pile'up}^r_{X_1, X_2}]{\text{pile'up}^l_{X_1, X_2}} T(X_1 \times X_2)$$

are equalized by both

$$X_1 \times T(X_2) \xrightarrow{\eta_{X_1} \times T(1_{X_2})} T(X_1) \times T(X_2)$$

and

$$T(X_1) \times X_2 \xrightarrow{T(1_{X_1}) \times \eta_{X_2}} T(X_1) \times T(X_2)$$

and their composition with these functions are equal to strength morphisms.

Using the diagram

$$\begin{array}{ccccc}
& & \eta_{T(X_1) \times X_2} & & \\
& \swarrow & & \searrow & \\
T(X_1) \times X_2 & \xrightarrow{T(1_{X_1}) \times \eta_{X_2}} & T(X_1) \times T(X_2) & \xrightarrow{\text{st}^r_{T(X_1), X_2}} & T(T(X_1) \times X_2) \\
\downarrow \text{st}^l_{X_1, X_2} & & \downarrow \text{st}^l_{T(X_1), X_2} & & \downarrow T(\text{st}^l_{X_1, X_2}) \\
T(X_1 \times X_2) & \xrightarrow{T(1_{X_1} \times \eta_{X_2})} & T(X_1 \times T(X_2)) & \xrightarrow{T(\text{st}^r_{X_1, X_2})} & T^2(X_1 \times X_2) \\
& \searrow & \swarrow T(\eta_{X_1 \times X_2}) & \nearrow & \downarrow \mu_{X_1 \times X_2} \\
& & \eta_{T(X_1 \times X_2)} & & \\
& \swarrow & & \searrow & \\
& & 1_{T(X_1 \times X_2)} & & T(X_1 \times X_2)
\end{array}$$

we shall show that

$$\text{pile'up}^r_{X_1, X_2} \circ (T(1_{X_1}) \times \eta_{X_2}) = \text{st}^l_{X_1, X_2} = \text{pile'up}^l_{X_1, X_2} \circ (T(1_{X_1}) \times \eta_{X_2}).$$

The other cases are symmetric. We have

$$\begin{aligned}
& \text{pile'up}^r_{X_1, X_2} \circ (T(1_{X_1}) \times \eta_{X_2}) = \text{ (def of pile'up}^r\text{)} \\
& = \mu_{X_1, X_2} \circ T(\text{st}^l_{X_1, X_2}) \circ \text{st}^r_{T(X_1), X_2} \circ (T(1_{X_1}) \times \eta_{X_2}) = \text{ (}\eta \text{ strong w.r.t. st}^r\text{)} \\
& = \mu_{X_1, X_2} \circ T(\text{st}^l_{X_1, X_2}) \circ \eta_{T(X_1) \times X_2} = \text{ (}\eta \text{ nat transf)}
\end{aligned}$$

$$\begin{aligned}
&= \mu_{X_1, X_2} \circ \eta_{T(X_1 \times X_2)} \circ \mathbf{st}^l_{X_1, X_2} = (T \text{ monad}) \\
&= \mathbf{st}^l_{X_1, X_2}
\end{aligned}$$

To show the remaining equation, we notice that we can continue the penultimate formula above as follows

$$\begin{aligned}
\mathbf{pile'up}^r_{X_1, X_2} \circ (T(1_{X_1}) \times \eta_{X_2}) &= \dots = \mu_{X_1, X_2} \circ \eta_{T(X_1 \times X_2)} \circ \mathbf{st}^l_{X_1, X_2} = (T \text{ monad}) \\
&= \mu_{X_1, X_2} \circ T(\eta_{X_1 \times X_2}) \circ \mathbf{st}^l_{X_1, X_2} = (\eta \text{ strong w.r.t. } \mathbf{st}^r) \\
&= \mu_{X_1, X_2} \circ T(\mathbf{st}^r_{X_1, X_2}) \circ T(1_{X_1} \times \eta_{X_2}) \circ \mathbf{st}^l_{X_1, X_2} = (\mathbf{st}^l \text{ nat transf}) \\
&= \mu_{X_1, X_2} \circ T(\mathbf{st}^r_{X_1, X_2}) \circ \mathbf{st}^l_{X_1, X_2} \circ T(1_{X_1} \times \eta_{X_2}) = (\text{def of } \mathbf{pile'up}^l) \\
&= \mathbf{pile'up}^l_{X_1, X_2} \circ (T(1_{X_1}) \times \eta_{X_2})
\end{aligned}$$

◇

Corollary 4.2 *The left and right CPS-operation on pairs where one element is continuized agree.*

Proof. The corollary states that, for any sets X, Y, Z and a function $f : X \times Y \rightarrow Z$, both morphisms

$$X \times T(Y) \xrightarrow{\eta_X \times 1} T(X) \times T(Y)$$

and

$$T(X) \times Y \xrightarrow{1 \times \eta_Y} T(X) \times T(Y)$$

equalize the pair of morphisms

$$\begin{array}{ccc}
T(X) \times T(Y) & \xrightarrow{\mathbf{CPS}^l(f)} & Z \\
& \xrightarrow{\mathbf{CPS}^r(f)} &
\end{array}$$

This immediately follows from the above lemma and the definition of **CPS**'es.

◇

Using binary pile'up operations, we can define eight ternary pile'up operation

$$T(X_1) \times T(X_2) \times T(X_3) \longrightarrow T(X_1 \times X_2 \times X_3)$$

out of the following diagram

$$\begin{array}{ccccc}
& & T(X_1) \times T(X_2) \times T(X_3) & & \\
& \swarrow & & \searrow & \\
\text{pile'up}^l \times 1 & & & & 1 \times \text{pile'up}^l \\
& \searrow & & \swarrow & \\
T^2(X_1 \times X_2) \times T(X_3) & & & & T(X_1) \times T^2(X_2 \times X_3) \\
& \swarrow & \searrow & \swarrow & \searrow \\
& & T^3(X_1 \times X_2 \times X_3) & &
\end{array}$$

However, both pile'up^l and pile'up^r operations are associative (Proposition 4.3 below) and hence only six of them are different, in general.

Proposition 4.3 *Both pile'up^l and pile'up^r operations are associative on any monad on Set .*

Proof. In fact, pile'up^l and pile'up^r are associative on any bi-strong monad on monoidal category. We shall show this fact for a monad T on Set with the canonical strength.

We need to show that

$$\text{pile'up}^r \circ (\text{pile'up}^r \times 1) = \text{pile'up}^r \circ (1 \times \text{pile'up}^r)$$

and

$$\text{pile'up}^l \circ (\text{pile'up}^l \times 1) = \text{pile'up}^l \circ (1 \times \text{pile'up}^l)$$

But as pile'up are mutually definable, either of these equalities implies easily the other. We shall show the latter equality. For sets X_1, X_2, X_3 , using all the assumptions, we have

$$\begin{aligned}
& \text{pile'up}^l_{X_1 \times X_2, X_3} \circ (\text{pile'up}^l_{X_1, X_2} \times 1_{T(X_3)}) = \\
& = \mu_{X_1, \times X_2 \times X_3} \circ T(\text{st}^r_{X_1 \times X_2, X_3}) \circ \underline{\text{st}^l_{X_1 \times X_2, T(X_3)}} \circ \\
& \circ (\underline{\mu_{X_1 \times X_2} \times T(1_{X_3})}) \circ (T(\text{st}^r_{X_1, X_2}) \times 1_{T(X_3)}) \circ (\text{st}^l_{X_1, T(X_2)} \times 1_{T(X_3)}) = \\
& = \mu_{X_1, \times X_2 \times X_3} \circ \underline{T(\text{st}^r_{X_1 \times X_2, X_3})} \circ \underline{\mu_{X_1 \times X_2 \times T(X_3)}} \circ T(\text{st}^l_{X_1 \times X_2, T(X_3)}) \circ
\end{aligned}$$

$$\begin{aligned}
& \circ \underline{\mathbf{st}^l_{T(X_1 \times X_2), T(X_3)}} \circ (T(\mathbf{st}^r_{X_1, X_2}) \times T(1_{X_3})) \circ (\mathbf{st}^l_{X_1, T(X_2)} \times 1_{T(X_3)}) = \\
& = \mu_{X_1, \times X_2 \times X_3} \circ \mu_{T(X_1 \times X_2 \times X_3)} \circ T^2(\mathbf{st}^r_{X_1 \times X_2, T(X_3)}) \circ T(\mathbf{st}^l_{X_1 \times X_2, T(X_3)}) \circ \\
& \quad \circ (T(\mathbf{st}^r_{X_1, X_2} \times 1_{X_3})) \circ \underline{\mathbf{st}^l_{T(X_1 \times X_2), T(X_3)}} \circ (\mathbf{st}^l_{X_1, T(X_2)} \times 1_{T(X_3)}) = \\
& = \mu_{X_1, \times X_2 \times X_3} \circ \mu_{T(X_1 \times X_2 \times X_3)} \circ T^2(\mathbf{st}^r_{X_1 \times X_2, T(X_3)}) \circ \underline{T(\mathbf{st}^l_{X_1 \times X_2, T(X_3)})} \circ \\
& \quad \circ \underline{(T(\mathbf{st}^r_{X_1, X_2} \times 1_{T(X_3)}))} \circ \mathbf{st}^l_{X_1, T(X_2) \times T(X_3)} = \\
& = \mu_{X_1, \times X_2 \times X_3} \circ \mu_{T(X_1 \times X_2 \times X_3)} \circ T^2(\mathbf{st}^r_{X_1 \times X_2, T(X_3)}) \circ T(\mathbf{st}^r_{X_1, X_2 \times T(X_3)}) \circ \\
& \quad \circ \underline{T(1_{X_1} \times \mathbf{st}^l_{X_2, T(X_3)})} \circ \mathbf{st}^l_{X_1, T(X_2) \times T(X_3)} = \\
& = \mu_{X_1, \times X_2 \times X_3} \circ \mu_{T(X_1 \times X_2 \times X_3)} \circ \underline{T^2(\mathbf{st}^r_{X_1 \times X_2, T(X_3)})} \circ T(\mathbf{st}^r_{X_1, X_2 \times T(X_3)}) \circ \\
& \quad \circ \mathbf{st}^l_{X_1, T(X_2 \times T(X_3))} \circ (T(1_{X_1}) \times \mathbf{st}^l_{X_2, T(X_3)}) = \\
& = \mu_{X_1, \times X_2 \times X_3} \circ \mu_{T(X_1 \times X_2 \times X_3)} \circ T^2(\mathbf{st}^r_{X_1, T(X_2 \times X_3)}) \circ \underline{T^2(1_{X_1} \times \mathbf{st}^r_{X_2, X_3})} \circ \\
& \quad \circ \underline{T(\mathbf{st}^r_{X_1, X_2 \times T(X_3)})} \circ \mathbf{st}^l_{X_1, T(X_2 \times T(X_3))} \circ (T(1_{X_1}) \times \mathbf{st}^l_{X_2, T(X_3)}) = \\
& = \mu_{X_1, \times X_2 \times X_3} \circ \mu_{T(X_1 \times X_2 \times X_3)} \circ T^2(\mathbf{st}^r_{X_1, T(X_2 \times X_3)}) \circ T(\mathbf{st}^r_{X_1, T(X_2 \times X_3)}) \circ \\
& \quad \circ \underline{T(1_{X_1} \times T(\mathbf{st}^r_{X_2, X_3}))} \circ \mathbf{st}^l_{X_1, T(X_2 \times T(X_3))} \circ (T(1_{X_1}) \times \mathbf{st}^l_{X_2, T(X_3)}) = \\
& = \mu_{X_1, \times X_2 \times X_3} \circ \underline{\mu_{T(X_1 \times X_2 \times X_3)} \circ T^2(\mathbf{st}^r_{X_1, T(X_2 \times X_3)})} \circ T(\mathbf{st}^r_{X_1, T(X_2 \times X_3)}) \circ \\
& \quad \circ \mathbf{st}^l_{X_1, T^2(X_2 \times X_3)} \circ (T(1_{X_1}) \times T(\mathbf{st}^r_{X_2, X_3})) \circ (T(1_{X_1}) \times \mathbf{st}^l_{X_2, T(X_3)}) = \\
& = \mu_{X_1, \times X_2 \times X_3} \circ T(\mathbf{st}^r_{X_1, X_2 \times X_3}) \circ \underline{T(1_{X_1} \times \mu_{X_2 \times X_3})} \circ
\end{aligned}$$

$$\begin{aligned}
& \circ \underline{\mathbf{st}^l_{X_1, T^2(X_2 \times X_3)}} \circ (T(1_{X_1}) \times T(\mathbf{st}^r_{X_2, X_3})) \circ (T(1_{X_1}) \times \mathbf{st}^l_{X_2, T(X_3)}) = \\
& = \mu_{X_1, \times X_2 \times X_3} \circ T(\mathbf{st}^r_{X_1, X_2 \times X_3}) \circ \mathbf{st}^l_{X_1, T(X_2 \times X_3)} \circ \\
& \circ (T(1_{X_1}) \times \mu_{X_2 \times X_3}) \circ (T(1_{X_1}) \times T(\mathbf{st}^r_{X_2, X_3})) \circ (T(1_{X_1}) \times \mathbf{st}^l_{X_2, T(X_3)}) = \\
& = \mathbf{pile'up}^l_{X_1, X_2 \times X_3} \circ (1_{T(X_3)} \times \mathbf{pile'up}^l_{X_2, X_3})
\end{aligned}$$

◇

4.3 Arity one: intransitive verbs

Proposition 4.4 *The semantics for sentences with intransitive verbs, as defined by the strategies A , B , and C , are equivalent.*

Proof. In case of a sentence with an intransitive verb the semantics are defined by morphisms \mathbf{strat}^1_A , \mathbf{strat}^1_B , and \mathbf{strat}^1_C . We need to show that they are equal. We have

$$\mathbf{strat}^1_A = \mathbf{mos}^l_X = \mathbf{strat}^1_B.$$

\mathbf{strat}^1_C is the composition of the following morphisms

$$\mathcal{C}(X) \times \mathcal{P}(X) \xrightarrow{1 \times \eta_{\mathcal{P}(X)}} \mathcal{C}(X) \times \mathcal{CP}(X) \xrightarrow{\mathbf{CPS}^l(\mathbf{eps}^r_X)} \mathcal{C}(\mathbf{t}) \xrightarrow{ev_{id_{\mathbf{t}}}} \mathbf{t}$$

Thus we need to show that this composition is equal to \mathbf{mos}^l_X . Consider the following diagram

$$\begin{array}{ccccc}
\mathcal{C}(X) \times \mathcal{P}(X) & \xrightarrow{\quad \mathbf{mos}^l_X \quad} & & & \mathbf{t} \\
\downarrow 1 \times \eta_{\mathcal{P}(X)} & \searrow \mathbf{st}^l & & \nearrow ev_{\mathbf{eps}^r_X} & \uparrow ev_{id_{\mathbf{t}}} \\
\mathcal{C}(X) \times \mathcal{CP}(X) & \xrightarrow{\quad \mathbf{pile'up}^l_X \quad} & \mathcal{C}(X \times \mathcal{P}(X)) & \xrightarrow{\quad \mathcal{C}(\mathbf{eps}^r_X) \quad} & \mathcal{C}(\mathbf{T})
\end{array}$$

The left triangle commutes, as a consequence of Lemma 4.1. To see that the mid triangle commutes, we take $M \in \mathcal{C}(X)$ and $h \in \mathcal{P}(X)$, and calculate

$$\begin{aligned}
& ev_{\mathbf{eps}^r_X} \circ \mathbf{st}^r(Q, h) = \\
& = ev_{\mathbf{eps}^r_X}(\lambda D : \mathcal{P}(X \times \mathcal{P}(X)) M(\lambda x : X D(x, h))) =
\end{aligned}$$

$$\begin{aligned}
&= M(\lambda x.X \mathbf{eps}^r_X(x, h)) = \\
&= M(\lambda x.X h(x)) = \\
&= N(h) = \mathbf{mos}^l(N, h).
\end{aligned}$$

Finally, to see that the right triangle commutes, we take $N \in \mathcal{C}(X \times \mathcal{P}(X))$ and calculate

$$\begin{aligned}
&ev_{id_t} \circ \mathcal{C}(\mathbf{eps}^r_X)(N) = \\
&= ev_{id_t}(\lambda c:\mathcal{P}(t) N(c \circ \mathbf{eps}^r_X)) = \\
&= N(\mathbf{eps}^r_X) = ev_{\mathbf{eps}^r_X}(N).
\end{aligned}$$

Thus the whole diagram commutes, and hence $\mathbf{strat}_C^1 = \mathbf{mos}^l_X$, as required.

◇

The above proof shows also the following technical lemma.

Lemma 4.5 *For any set X , the diagram*

$$\begin{array}{ccc}
\mathcal{C}(X) \times \mathcal{P}(X) & \xrightarrow{\mathbf{mos}^l_X} & \mathbf{t} \\
\mathbf{st}^l \downarrow & & \uparrow ev_{id_t} \\
\mathcal{C}(X \times \mathcal{P}(X)) & \xrightarrow{\mathcal{C}(\mathbf{eps}^r_X)} & \mathcal{C}(\mathbf{t})
\end{array}$$

commutes.

4.4 Arity two: transitive verbs

Proposition 4.6 *The semantics for sentences with transitive verbs, as defined by the strategies A , B , and C , are equivalent. They provide two asymmetric readings of the sentence.*

Proof. In case of sentences with transitive verbs the semantics are defined by morphisms $\mathbf{strat}_A^{2,\sigma}$, $\mathbf{strat}_B^{2,\sigma}$, and $\mathbf{strat}_C^{2,\varepsilon}$, with $\sigma \in S_2 = \{id_2, \tau\}$ and $\varepsilon \in \{l, r\}$. We need to show the equalities

$$\mathbf{strat}_A^{2,\sigma} = \mathbf{strat}_B^{2,\sigma},$$

for $\sigma \in S_2$, and

$$\mathbf{strat}_B^{2,id_2} = \mathbf{strat}_C^{2,l}, \quad \mathbf{strat}_B^{2,\tau} = \mathbf{strat}_C^{2,r}.$$

To show the first equality, with $Q_1 \in \mathcal{C}(X_1)$, $Q_2 \in \mathcal{C}(X_2)$, and $P \in \mathcal{P}(X_1 \times X_2)$, we have

$$\begin{aligned}
& \mathbf{strat}_A^{2,\sigma}(Q_1, Q_2, P) = \\
& = \mathbf{mos}^l_{X_{\sigma(1)}}(Q_{\sigma(1)}, \mathbf{mos}^l_{X_{\sigma(2)}}(Q_{\sigma(2)}, P)) = \\
& = \mathbf{mos}^l_{X_{\sigma(1)}}(Q_{\sigma(1)}, \lambda x_{\sigma(1):X_{\sigma(1)}}.Q_{\sigma(2)}(\lambda x_{\sigma(2):X_{\sigma(2)}}.P(x_1, x_2))) = \\
& = Q_{\sigma(1)}(\lambda x_{\sigma(1):X_{\sigma(1)}}.Q_{\sigma(2)}(\lambda x_{\sigma(2):X_{\sigma(2)}}.P(x_1, x_2))) = \\
& = Q_{\sigma(1)}(\lambda x_{\sigma(1):X_{\sigma(1)}}.Q_{\sigma(2)}(\lambda x_{\sigma(2):X_{\sigma(2)}}.P(\pi_{\sigma^{-1}}(x_{\sigma(1)}, x_{\sigma(2)}))) = \\
& = \mathbf{pile'up}^l(Q_{\sigma(1)}, Q_{\sigma(2)})(P \circ \pi^{-1}) = \\
& = \mathcal{C}(\pi_{\sigma^{-1}})(\mathbf{pile'up}^l(Q_{\sigma(1)}, Q_{\sigma(2)}))(P) = \\
& = \mathbf{strat}_B^{2,\sigma}(Q_1, Q_2, P)
\end{aligned}$$

To show the remaining two equalities, let us first note that if either $\sigma = id_2$ and $\varepsilon = l$ or $\sigma = \tau$ and $\varepsilon = r$, we have

$$\mathbf{pile'up}^\varepsilon = \mathcal{C}(\pi_{\sigma^{-1}}) \circ \mathbf{pile'up}^l \circ \pi_\sigma.$$

Thus we shall assume the above equation relating σ with ε , and, with $Q_1 \in \mathcal{C}(X_1)$, $Q_2 \in \mathcal{C}(X_2)$, and $P \in \mathcal{P}(X_1 \times X_2)$, we obtain (the diagram illustrating these calculations would be too big to fit a page but the reader is encouraged to draw one)

$$\begin{aligned}
& \mathbf{strat}_C^{2,\varepsilon} = \\
& = ev_{id_t} \circ \mathbf{CPS}^\varepsilon(\mathbf{eps}^r_{X_1}) \circ (1 \times \mathbf{CPS}^?(\mathbf{eps}^r_{X_2})) \circ (1 \times 1 \times \eta_{\mathcal{P}(X_1 \times X_2)}) = \\
& = ev_{id_t} \circ \mathcal{C}(\mathbf{eps}^r_{X_1}) \circ \mathbf{pile'up}^\varepsilon \circ (\mathcal{C}(1) \times \mathcal{C}(\mathbf{eps}^r_{X_2})) \circ (1 \times \mathbf{pile'up}^?) \circ (1 \times 1 \times \eta) = \\
& = ev_{id_t} \circ \mathcal{C}(\mathbf{eps}^r_{X_1}) \circ \mathcal{C}(1 \times \mathbf{eps}^r_{X_2}) \circ \mathbf{pile'up}^\varepsilon \circ (1 \times \mathbf{pile'up}^?) \circ (1 \times 1 \times \eta) = \\
& = ev_{id_t} \circ \mathcal{C}(\mathbf{eps}^r_{X_1 \times X_2}) \circ \mathbf{pile'up}^\varepsilon \circ (1 \times \mathbf{pile'up}^?) \circ (1 \times 1 \times \eta) = \\
& = ev_{id_t} \circ \mathcal{C}(\mathbf{eps}^r_{X_1 \times X_2}) \circ \mathbf{pile'up}^? \circ (\mathbf{pile'up}^\varepsilon \times 1) \circ (1 \times 1 \times \eta) = \\
& = ev_{id_t} \circ \mathcal{C}(\mathbf{eps}^r_{X_1 \times X_2}) \circ \mathbf{pile'up}^? \circ (1 \times \eta) \circ (1 \times \mathbf{pile'up}^\varepsilon) = \\
& = ev_{id_t} \circ \mathcal{C}(\mathbf{eps}^r_{X_1 \times X_2}) \circ \mathbf{st}^l \circ (1 \times \mathbf{pile'up}^\varepsilon) = \\
& = ev_{id_t} \circ \mathcal{C}(\mathbf{eps}^r_{X_1 \times X_2}) \circ \mathbf{st}^l \circ (\mathcal{C}(\pi_{\sigma^{-1}}) \times 1) \circ (\mathbf{pile'up}^l \times 1) \circ (\pi_\sigma \times 1) = \\
& = \mathbf{mos}^l_{X_1 \times X_2} \circ (\mathcal{C}(\pi_{\sigma^{-1}}) \times 1) \circ (\mathbf{pile'up}^l \times 1) \circ (\pi_\sigma \times 1) =
\end{aligned}$$

$$= \mathbf{strat}_B^{2,\sigma}$$

In the above calculations we used: the definition of **CPS**'es, naturality of **pile'up**^ε, relations between **eps** morphisms, associativity of **pile'up**^ε (Proposition 4.3), properties of product morphisms, pile'up lemma, and finally Lemma 4.5.

Here and below **CPS**[?], **pile'up**[?] stands for either **CPS**^l, **pile'up**^l or **CPS**^r, **pile'up**^r whatever is more convenient at the moment as it does not influence the end result. ◇

4.5 Arity three: ditransitive verbs

Proposition 4.7 *The semantics for sentences with ditransitive verbs, as defined by the strategies A , B , are equivalent. They provide six asymmetric readings of the sentence.*

Proof. In case of sentences with ditransitive verbs the semantics are defined by morphisms $\mathbf{strat}_A^{3,\sigma}$, $\mathbf{strat}_B^{3,\sigma}$, and $\mathbf{strat}_C^{2,\varepsilon}$, with $\sigma \in S_3$ and $\varepsilon \in \{l, r\}$. We need to show the equalities

$$\mathbf{strat}_A^{3,\sigma} = \mathbf{strat}_B^{3,\sigma},$$

for $\sigma \in S_3$.

The calculations are similar to those for transitive verbs. We present them for completeness. With $Q_1 \in \mathcal{C}(X_1)$, $Q_2 \in \mathcal{C}(X_2)$, $Q_3 \in \mathcal{C}(X_3)$, and $P \in \mathcal{P}(X_1 \times X_2 \times X_3)$, we have

$$\begin{aligned} \mathbf{strat}_A^{3,\sigma}(Q_1, Q_2, Q_3, P) &= \\ &= \mathbf{mos}^l_{X_{\sigma(1)}}(Q_{\sigma(1)}, \mathbf{mos}^l_{X_{\sigma(2)}}(Q_{\sigma(2)}, \mathbf{mos}^l_{X_{\sigma(3)}}(Q_{\sigma(3)}, P)) = \\ &= Q_{\sigma(1)}(\lambda x_{\sigma(1):X_{\sigma(1)}}. Q_{\sigma(2)}(\lambda x_{\sigma(2):X_{\sigma(2)}}. Q_{\sigma(3)}(\lambda x_{\sigma(3):X_{\sigma(3)}}. P(x_1, x_2, x_3))) = \\ &= Q_{\sigma(1)}(\lambda x_{\sigma(1):X_{\sigma(1)}}. Q_{\sigma(2)}(\lambda x_{\sigma(2):X_{\sigma(2)}}. Q_{\sigma(3)}(\lambda x_{\sigma(3):X_{\sigma(3)}}. P(\pi_{\sigma^{-1}}(x_{\sigma(1)}, x_{\sigma(2)}, x_{\sigma(3)})))) = \\ &= \mathbf{pile'up}^l(Q_{\sigma(1)}, \mathbf{pile'up}^l(Q_{\sigma(2)}, Q_{\sigma(3)}))(P \circ \pi_{\sigma^{-1}}) = \\ &= \mathcal{C}(\pi_{\sigma^{-1}})(\mathbf{pile'up}^l(Q_{\sigma(1)}, \mathbf{pile'up}^l(Q_{\sigma(2)}, Q_{\sigma(3)})))(P) = \\ &= \mathbf{strat}_B^{2,\sigma}(Q_1, Q_2, Q_3, P) \end{aligned}$$

as required. ◇

Proposition 4.8 *The semantics for sentences with ditransitive verbs, as defined by the strategy C, provides four asymmetric readings of the sentence such that QP in subject position can be placed either first or last only. Thus they correspond to four out of six readings accounted for by strategies A and B.*

Proof. In case of sentences with ditransitive verbs the semantics, according to strategies B and C, are defined by morphisms $\mathbf{strat}_B^{3,\sigma}$, $\mathbf{strat}_C^{3,\varepsilon,\varepsilon'}$, respectively. As we shall show, these morphisms are equal whenever $\sigma \in S_3$ is related to the pair $\langle \varepsilon', \varepsilon \rangle \in \{l, r\}^2$ via relation

$$\mathbf{pile}'\mathbf{up}^{\varepsilon'} \circ (1 \times \mathbf{pile}'\mathbf{up}^{\varepsilon}) = \mathcal{C}(\pi_{\sigma^{-1}}) \circ \mathbf{pile}'\mathbf{up}^l \circ (1 \times \mathbf{pile}'\mathbf{up}^l) \circ \pi_{\sigma}$$

As $\mathbf{pile}'\mathbf{up}^l$ leaves the order intact and $\mathbf{pile}'\mathbf{up}^r$ swaps the order, we can see that we have the following correspondence

σ	$\langle \varepsilon', \varepsilon \rangle$
(1, 2, 3)	$\langle l, l \rangle$
(1, 3, 2)	$\langle l, r \rangle$
(2, 3, 1)	$\langle r, l \rangle$
(3, 2, 1)	$\langle r, r \rangle$
(2, 1, 3)	—
(3, 1, 2)	—

Thus we shall assume the σ is related to the pair $\langle \varepsilon, \varepsilon' \rangle$, and, with $Q_1 \in \mathcal{C}(X_1)$, $Q_2 \in \mathcal{C}(X_2)$, $Q_3 \in \mathcal{C}(X_3)$, and $P \in \mathcal{P}(X_1 \times X_2 \times X_3)$, we obtain (the diagram illustrating these calculations would be again too big to fit a page but the reader is encouraged to draw one)

$$\begin{aligned}
& \mathbf{strat}_C^{3,\varepsilon',\varepsilon} = \\
& = ev_{id_t} \circ \mathbf{CPS}^{\varepsilon'}(\mathbf{eps}^r_{X_1}) \circ (1 \times \mathbf{CPS}^{\varepsilon}(\mathbf{eps}^r_{X_2})) \circ \\
& \quad \circ (1 \times 1 \times \mathbf{CPS}^{\varepsilon'}(\mathbf{eps}^r_{X_3})) \circ (1 \times 1 \times 1 \times \eta) = \\
& = ev_{id_t} \circ \mathcal{C}(\mathbf{eps}^r_{X_1}) \circ \mathbf{pile}'\mathbf{up}^{\varepsilon'} \circ (\mathcal{C}(1) \times \mathcal{C}(\mathbf{eps}^r_{X_2})) \circ (1 \times \mathbf{pile}'\mathbf{up}^{\varepsilon}) \circ \\
& \quad \circ (\mathcal{C}(1) \times \mathcal{C}(1) \times \mathcal{C}(\mathbf{eps}^r_{X_3})) \circ (1 \times 1 \times \mathbf{pile}'\mathbf{up}^{\varepsilon'}) \circ (1 \times 1 \times 1 \times \eta) =
\end{aligned}$$

$$\begin{aligned}
&= ev_{id_t} \circ \mathcal{C}(\mathbf{eps}^r_{X_1}) \circ (\mathcal{C}(1 \times \mathbf{eps}^r_{X_2})) \circ \mathbf{pile}'\mathbf{up}^{\varepsilon'} \circ (\mathcal{C}(1) \times \mathcal{C}(1 \times \mathbf{eps}^r_{X_3})) \circ \\
&\quad \circ (1 \times \mathbf{pile}'\mathbf{up}^\varepsilon) \circ (1 \times 1 \times \mathbf{pile}'\mathbf{up}^?) \circ (1 \times 1 \times 1 \times \eta) = \\
&= ev_{id_t} \circ \mathcal{C}(\mathbf{eps}^r_{X_1}) \circ (\mathcal{C}(1 \times \mathbf{eps}^r_{X_2})) \circ (\mathcal{C}(1 \times 1 \times \mathbf{eps}^r_{X_3})) \circ \mathbf{pile}'\mathbf{up}^{\varepsilon'} \circ \\
&\quad \circ (1 \times \mathbf{pile}'\mathbf{up}^\varepsilon) \circ (1 \times 1 \times \mathbf{pile}'\mathbf{up}^?) \circ (1 \times 1 \times 1 \times \eta) = \\
&\quad = ev_{id_t} \circ \mathcal{C}(\mathbf{eps}^r_{X_1 \times X_2 \times X_3}) \circ \mathbf{pile}'\mathbf{up}^{\varepsilon'} \circ \\
&\quad \circ (1 \times \mathbf{pile}'\mathbf{up}^\varepsilon) \circ (1 \times 1 \times \mathbf{pile}'\mathbf{up}^?) \circ (1 \times 1 \times 1 \times \eta) = \\
&\quad = ev_{id_t} \circ \mathcal{C}(\mathbf{eps}^r_{X_1 \times X_2 \times X_3}) \circ \mathbf{pile}'\mathbf{up}^{\varepsilon'} \circ \\
&\quad \circ (1 \times \mathbf{pile}'\mathbf{up}^\varepsilon) \circ (1 \times 1 \times \mathbf{pile}'\mathbf{up}^?) \circ (1 \times 1 \times 1 \times \eta) = \\
&\quad = ev_{id_t} \circ \mathcal{C}(\mathbf{eps}^r_{X_1 \times X_2 \times X_3}) \circ \mathbf{pile}'\mathbf{up}^{\varepsilon'} \circ \\
&\quad \circ (1 \times \mathbf{pile}'\mathbf{up}^?) \circ (1 \times \mathbf{pile}'\mathbf{up}^\varepsilon \times 1) \circ (1 \times 1 \times 1 \times \eta) = \\
&\quad = ev_{id_t} \circ \mathcal{C}(\mathbf{eps}^r_{X_1 \times X_2 \times X_3}) \circ \mathbf{pile}'\mathbf{up}^? \circ \\
&\quad \circ (\mathbf{pile}'\mathbf{up}^{\varepsilon'} \times 1) \circ (1 \times \mathbf{pile}'\mathbf{up}^\varepsilon \times 1) \circ (1 \times 1 \times 1 \times \eta) \stackrel{*}{=} \\
&\quad \stackrel{*}{=} ev_{id_t} \circ \mathcal{C}(\mathbf{eps}^r_{X_1 \times X_2 \times X_3}) \circ \mathbf{pile}'\mathbf{up}^? \circ \\
&\quad \circ (\mathcal{C}(\pi_{\sigma^{-1}}) \times \mathcal{C}(1)) \circ (\mathbf{pile}'\mathbf{up}^l \times 1) \circ (1 \times \mathbf{pile}'\mathbf{up}^l \times 1) \circ (\pi_\sigma \times 1) \circ (1 \times 1 \times 1 \times \eta) = \\
&\quad = ev_{id_t} \circ \mathcal{C}(\mathbf{eps}^r_{X_1 \times X_2 \times X_3}) \circ (\mathcal{C}(\pi_{\sigma^{-1}} \times 1) \circ \mathbf{pile}'\mathbf{up}^? \circ \\
&\quad \circ (\mathbf{pile}'\mathbf{up}^l \times 1) \circ (1 \times \mathbf{pile}'\mathbf{up}^l \times 1) \circ (\pi_\sigma \times 1) \circ (1 \times 1 \times 1 \times \eta) = \\
&\quad = ev_{id_t} \circ \mathcal{C}(\mathbf{eps}^r_{X_1 \times X_2 \times X_3}) \circ (\mathcal{C}(\pi_{\sigma^{-1}} \times 1) \circ \mathbf{pile}'\mathbf{up}^? \circ \\
&\quad \circ (\mathbf{pile}'\mathbf{up}^l \times 1) \circ (1 \times \mathbf{pile}'\mathbf{up}^l \times 1) \circ (1 \times 1 \times 1 \times \eta) \circ (\pi_\sigma \times 1) =
\end{aligned}$$

$$\begin{aligned}
&= ev_{id_t} \circ \mathcal{C}(\mathbf{eps}^r_{X_1 \times X_2 \times X_3}) \circ (\mathcal{C}(\pi_{\sigma^{-1}} \times 1) \circ \mathbf{pile}'\mathbf{up}^? \circ \\
&\circ (\mathbf{pile}'\mathbf{up}^l \times 1) \circ (1 \times \times \eta) \circ (1 \times \mathbf{pile}'\mathbf{up}^l \times 1) \circ (\pi_\sigma \times 1) = \\
&= ev_{id_t} \circ \mathcal{C}(\mathbf{eps}^r_{X_1 \times X_2 \times X_3}) \circ (\mathcal{C}(\pi_{\sigma^{-1}} \times 1) \circ \mathbf{pile}'\mathbf{up}^? \circ \\
&= \circ (\mathbf{pile}'\mathbf{up}^l \times 1) \circ (1 \times 1 \times \eta) \circ (1 \times \mathbf{pile}'\mathbf{up}^l \times 1) \circ (\pi_\sigma \times 1) = \\
&= ev_{id_t} \circ \mathcal{C}(\mathbf{eps}^r_{X_1 \times X_2 \times X_3}) \circ (\mathcal{C}(\pi_{\sigma^{-1}} \times 1) \circ \mathbf{pile}'\mathbf{up}^? \circ \\
&\circ (1 \times \eta) \circ (\mathbf{pile}'\mathbf{up}^l \times 1) \circ (1 \times \mathbf{pile}'\mathbf{up}^l \times 1) \circ (\pi_\sigma \times 1) = \\
&= ev_{id_t} \circ \mathcal{C}(\mathbf{eps}^r_{X_1 \times X_2 \times X_3}) \circ (\mathcal{C}(\pi_{\sigma^{-1}} \times 1) \circ \mathbf{st}^{l^l} \circ \\
&\circ (\mathbf{pile}'\mathbf{up}^l \times 1) \circ (1 \times \mathbf{pile}'\mathbf{up}^l \times 1) \circ (\pi_\sigma \times 1) = \\
&= ev_{id_t} \circ \mathcal{C}(\mathbf{eps}^r_{X_1 \times X_2 \times X_3}) \circ \mathbf{st}^{l^l} \circ (\mathcal{C}(\pi_{\sigma^{-1}}) \times \mathcal{C}(1)) \circ \\
&\circ (\mathbf{pile}'\mathbf{up}^l \times 1) \circ (1 \times \mathbf{pile}'\mathbf{up}^l \times 1) \circ (\pi_\sigma \times 1) = \\
&= \mathbf{mos}^l_{X_1 \times X_2 \times X_3} \circ (\mathcal{C}(\pi_{\sigma^{-1}}) \times \mathcal{C}(1)) \circ \\
&\circ (\mathbf{pile}'\mathbf{up}^l \times 1) \circ (1 \times \mathbf{pile}'\mathbf{up}^l \times 1) \circ (\pi_\sigma \times 1) = \\
&= \mathbf{strat}^{3,\sigma}_B
\end{aligned}$$

In the above calculations we used: the definition of **CPS**'es, naturality of **pile'up**'s (four times in three non-consecutive steps!), relations between **eps** morphisms, associativity of **pile'up**'s (Proposition 4.3), relations between σ and $\langle \varepsilon', \varepsilon \rangle$, properties of product morphisms (three consecutive steps), pile'up lemma, naturality of strength, and finally Lemma 4.5. \diamond

References

- [1] Barker, C.: Continuations and the nature of quantification. *Natural Language Semantics* 10, 211-242 (2002).
- [2] Barker, C., Shan, C.c.: *Continuations and Natural Language*. Oxford University Press (2014).
- [3] Barwise, J., Cooper, R.: Generalized Quantifiers and Natural Language. *Linguistics & Philosophy* 4, 159-219 (1981).
- [4] Bekki, D., Assai, K.: Representing Covert Movements by Delimited Continuations In Nakakoji, K., Murakami, Y., McCready, E. (eds.) *New Frontiers in Artificial Intelligence, JSAI-isAI, LNAI 6284*, 161-180, (2010).
- [5] Benthem, J.: Polyadic quantifiers. *Linguistics & Philosophy* 12, 437-464 (1989).
- [6] Cooper, R.: *Quantification and Syntactic Structure*. Dordrecht: Reidel (1983).
- [7] de Groote, P.: Type raising, continuations, and classical logic. In van Rooy, R., Stokhof, M. (eds.), *Proceedings of the 13th Amsterdam Colloquium, Institute for Logic, Language and Computation, Universiteit van Amsterdam 97-101* (2001).
- [8] Eilenberg, S., Kelly, M.: *Closed Categories*. *Proc. Conf. Categorical Algebra* (La Jolla 1965), Springer-Verlag 1966.
- [9] Keenan, E. L.: Unreducible n-ary quantifiers in natural language. In Grdenfors P. (ed.), *Generalized Quantifier: Linguistic and Logical Approaches*, Reidel Dordrecht. 109-150 (1987).
- [10] Keenan, E. L.: Beyond the Frege Boundary. *Linguistics & Philosophy* 15, 199-221 (1992).
- [11] Kiselyov, O., Shan, C.c.: Continuation Hierarchy and Quantifier Scope. In McCready, E., Yabushita, K., Yoshimoto, K. (eds.), *Formal Approaches to Semantics and Pragmatics: Japanese and Beyond. Studies in Linguistics and Philosophy*, Springer Netherlands, 105-134 (2014).
- [12] Kock, A.: Monads on symmetric monoidal closed categories. *Arch. Math. (Basel)*, 21:110 (1970).

- [13] Kock, A.: Closed categories generated by commutative monads. *J. Austral. Math. Soc.*, 12:405-424 (1971).
- [14] Kock, A.: Strong functors and monoidal monads. *Arch. Math. (Basel)*, 23:113-120 (1972).
- [15] Lindström, P.: First-order predicate logic with generalized quantifiers. *Theoria* 32, 186-95.(1966).
- [16] May, R.: *The Grammar of Quantification*. PhD dissertation, MIT (1977).
- [17] May, R.: *Logical Form: Its Structure and Derivation*. MIT Press (1985).
- [18] Moggi, E.: The notion of computation and monads. *Information And Computation* 93:55-92 (1991).
- [19] Montague, R.: The proper treatment of quantification in ordinary English. In: Thomason, R. (ed.), *Formal Philosophy: Selected Papers of Richard Montague*. New Haven and London: Yale University Press, 247-271 (1974).
- [20] Mostowski, A.: On a generalization of quantifiers. *Fundamenta Mathematicae* 44, 12-36 (1957).
- [21] Szabolcsi, A.: *Quantification*. Cambridge University Press, Cambridge (2010).
- [22] Zawadowski, M.: Formalization of the feature system in terms of pre-orders. In Bellert, I, *Feature System for Quantification Structures in Natural Language*, Foris Dordrecht, 155-175 (1989).